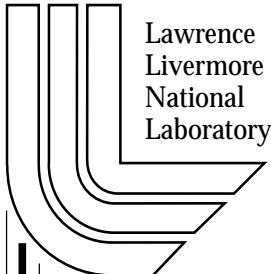


# **MCAPM-C Generator and Collision Routine (gen2000/bang2000) Documentation**

*Patrick S. Brantley, Chris A. Hagmann, Jim A. Rathkopf*

November 22, 2000

*U.S. Department of Energy*



## **DISCLAIMER**

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

## Abstract

This report documents the C version of the **MCAPM** (Monte Carlo All Particle Method) generator and collision physics library of subroutines. The Monte Carlo data generator (`gen2000`) reads cross sections and distributions that describe in-flight reactions from a binary library and creates an internal data library. The collision routines (`bang2000`) use this internal library to perform the physics of interaction of particles with the background material. Particles modeled with **MCAPM-C** are neutrons, charged particles (p, d, t,  $^3\text{He}$ , and  $\alpha$ ), and  $\gamma$ 's. **MCAPM-C** is written in (nearly) standard C, and versions exist for Sun Solaris, Compaq Unix, IBM AIX, SGI Irix, and Linux. The library and its data files are installed on LC's Compass, TC, Forest, Blue, and Sky machines. This report describes the contents and format of the library, physics assumptions made, and the interface to the library's subroutines.

## Table of Contents

|  |     |
|--|-----|
| DISCLAIMER.....                                    | i   |
| Abstract.....                                      | ii  |
| Table of Contents.....                             | iii |
| List of Tables .....                               | iv  |
| Introduction.....                                  | 1   |
| Acknowledgments.....                               | 1   |
| User's Manual .....                                | 2   |
| Projectiles or Products?.....                      | 2   |
| Cross Section Data.....                            | 2   |
| Projectile Multiplicities .....                    | 4   |
| Projectile Characteristic Distributions .....      | 4   |
| Available Isotopes .....                           | 4   |
| Depletion/Accretion Data.....                      | 5   |
| Kinematics Physics.....                            | 5   |
| Kinematics Type 1: Relativistic Two-Body .....     | 5   |
| Kinematics Type 2: Energy/Angle Uncorrelated ..... | 5   |
| Kinematics Type 3: Energy/Angle Correlated .....   | 6   |
| Kinematics Type 4: Decay .....                     | 6   |
| Kinematics Type 5: Cluster .....                   | 6   |
| Kinematics Type 6: Coherent Scattering.....        | 6   |
| Kinematics Type 7: Incoherent Scattering.....      | 7   |
| Definitions .....                                  | 8   |
| Client Code Programmer's Manual .....              | 16  |
| Usage Overview .....                               | 16  |
| Access Routines .....                              | 17  |
| Customization Routines .....                       | 20  |
| Non-Analog Physics.....                            | 21  |
| Sourcing Modes-Reaction and Emission.....          | 21  |
| Group Collapse.....                                | 21  |
| Energy Deposition.....                             | 22  |
| Random Number Generator .....                      | 22  |
| Library Access.....                                | 22  |
| Future Improvements .....                          | 22  |
| gen2000/bang2000 Calling Sequences.....            | 23  |
| Library Previewer .....                            | 23  |
| Generator.....                                     | 23  |
| Access Routines.....                               | 25  |
| Collision Routines .....                           | 33  |
| Client Customization Routines .....                | 39  |
| MCAPM Include File .....                           | 43  |
| Index.....   | 48  |
| Document Revision History .....                    | 50  |

## List of Tables

### User's Manual:

|  |    |
|--|----|
| Projectile Particles and APM Library File Names.....       | 8  |
| Energy Group Boundaries (neutron library) .....            | 9  |
| Energy Group Boundaries (charged particle libraries) ..... | 10 |
| Energy Points (gamma library) .....                        | 11 |
| Available Targets (neutron library) .....                  | 12 |
| Available Targets (charged particle libraries) .....       | 13 |
| Available Targets (gamma library) .....                    | 14 |
| Reaction IDs (C-numbers) .....                             | 15 |
| Code Units.....  | 15 |

### Client Code Programmer's Manual:

|  |    |
|--|----|
| Data Available from gen2000 Access Routines.....     | 18 |
| Data Accessible by MCAPM Customization Routines..... | 20 |

## Introduction

The Monte Carlo All Particle Method (**MCAPM**) generator and collision package provides a means for client codes to access the All Particle Method data base and simulate in-flight reaction physics. All particle types are coupled through the collision physics. Neutrons incident on a target may, as determined by the data, produce any of the particles treated in the library: neutrons, charged particles (protons, deuterons, tritons,  $^3\text{He}$ , and alpha), and gammas. Tritons, for example, may similarly produce any of these same particles. The data libraries are organized by incident particle. Libraries for all particle types have identical formats and access routines.

This report consists of three major sections. The first section contains information of interest to users of client codes and may be extracted from this report by client code developers and incorporated into their own documentation. The “User’s Manual” includes description of physics used in the collision routines and lists of library characteristics such as incident particles available, energy group boundaries, available targets, and reaction IDs. All of these lists (except for reaction IDs) may be extracted directly from the library through the use of provided access routines.

The second section provides information needed by client code developers interested in using **MCAPM**. The “Client Code Programmer’s Manual” describes routines that may be replaced by the programmer, routines that must be provided by the programmer; calling sequences to access routines that allow the client code to extract specific data, such as cross sections and yields, with no knowledge of the details of the library layout; and calling sequences to the generator (gen2000) and the collision routine (bang2000) and parts of the collision routines themselves. Options that allow non-analog physics are described.

The third section is the “**MCAPM** Programmer’s Manual” which provides detailed description of the library layout and descriptions of all **MCAPM** routines and include files in the form of the prologue of each subroutine. *This section is not yet completed.*

## Acknowledgments

This document is an updated version of an unpublished report by Jim Rathkopf, who wrote the original MCAPM library.

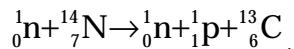
## User's Manual

The Monte Carlo All Particle Method (**MCAPM**) generator, `gen2000`, reads the data files listed in the table “Projectile Particles and APM Library File Names.” The files are in portable binary data (pdb) format and are generated with the **MCFGEN** package. The data files are organized by incident particle. The file `mcf2.pdb`, for example, describes the interaction of protons with the background medium and provides information describing the characteristics of secondary particles, be they neutrons, protons, gammas, or any other of the seven particles considered. The term *secondary particle* refers to a particle emerging from a reaction. Even if the emerging particle is the same one causing the reaction, for example in elastic scattering, it is still considered a secondary particle.

Four major types of data are contained in these data files: cross sections, projectile multiplicities, projectile characteristic distributions (angle and energy distributions), and depletion/accretion information.

### Projectiles or Products?

The mass, momentum, and energy brought into the reaction by the incident particle and target isotope are transferred to the emerging particles. Although all particles, in reality, share the available momentum and energy, only the momentum and energy of *projectiles* listed in the table “Projectile Particles and APM Library File Names” can be explicitly calculated by **MCAPM**'s `bang2000`. Further, `bang2000` will consider to be projectiles only those particles listed in the table that are selected by the client code. All other objects emerging from a reaction are considered *products*. Consider, for example, the (n,n'p) reaction off  $^{14}\text{N}$ :



If the client code is tracking neutrons and protons, the result of this reaction is the production of two projectiles, neutron (y=1) and proton (y=2), and one product,  $^{13}\text{C}$  (6013). If the client code is tracking only neutrons, this reaction produces one projectile, neutron (y=1), and two products,  $^1\text{H}$  (1001) and  $^{13}\text{C}$  (6013).

Although the momentum and energy is not explicitly calculated on a reaction-by-reaction basis for reaction products, conservation of energy is maintained in the average, statistical sense through the use of energy deposition, a value determined by `gen2000` as a function of incident particle energy, target isotope, reaction, and tracked particles as specified by the client code. The client code deposits this energy to the background medium. No mechanism is provided for correcting for the momentum carried away by products.

### Cross Section Data

The method of tabulating cross sections depends on the incident particle type. In the case of neutrons and charged particles, the tabulated cross sections have been averaged with a flat parti-

cle spectrum over energy intervals or *groups*. The value of the cross section is taken to be constant over a group. The neutron multi-group cross sections span the energy range between  $1.3 \times 10^{-3}$  eV to 20 MeV in 175 groups. Charged particle multi-group cross sections are tabulated over 63 groups ranging from  $1.0 \times 10^{-4}$  eV to 20 MeV. Although the cross sections are considered multi-group, Monte Carlo particles possess a discrete value of energy allowing kinematics to be performed on a continuous energy basis. This Plechaty Hybrid scheme allows exploitation of the strengths of both the pure continuous energy method and the pure multi-group method: a small cross section data base and accurate kinematics.

Gamma cross sections are tabulated at 176 energy points. Logarithmic interpolation is used to find the cross section value over the interval between tabulated points. Care must be taken with logarithmic interpolation as summed cross sections cannot be interpolated via log-log even though its constituents can.

Tables below give the energy group boundaries or energy points present in the libraries. Note that these energy values (including the number of values) are not *hard-wired* into the **MCAPM** package but are determined by **MCAPM** by reading the data libraries. This data-driven feature allows flexibility in library characteristics without producing new code.

Special consideration must be given the case where a reaction threshold energy (i.e., the incident particle energy at which the total kinetic energy of the exiting products is exactly zero) falls within a energy group rather than at a group boundary. An incident particle with energy below this threshold does not have sufficient energy to instigate such a reaction. The group cross section, however, is non-zero and constant for all incident particles with energies within the group. **MCAPM** handles this situation by prohibiting reactions below threshold in `bang2000` and preserves the reaction rate by adjusting the threshold group cross section in `gen2000` to compensate for the prohibited reactions. Consider a neutron experiencing an inelastic scatter off the 0.148 MeV level of  $^{238}\text{U}$ . The continuous energy cross section rises linearly from 0.0 b at the threshold of 148.62 keV to 0.15 b at the second tabulated point at 400 keV. The multigroup cross section given in `mcf1.pdb` is  $6.467 \times 10^{-3}$  b in the threshold group which extends from 130.68 to 181.95 keV. The `mcf1.pdb` value is found by calculating the integral

$$\sigma_g = \frac{\int_{E_g}^{E_{g+1}} \sigma(E) dE}{\int_{E_g}^{E_{g+1}} dE} = \frac{\int_{E_{thresh}}^{E_{g+1}} \sigma(E) dE}{E_{g+1} - E_g}.$$

That is,

$$\sigma_g (E_{g+1} - E_g) = \int_{E_{thresh}}^{E_{g+1}} \sigma(E) dE.$$

Because `bang2000` discards any reactions occurring for neutrons with energies below the threshold, the reaction rate must be realized over a narrower interval:

$$\sigma_{g,adj} (E_{g+1} - E_{thresh}) = \int_{E_{thresh}}^{E_{g+1}} \sigma(E) dE.$$

This adjusted group cross section is calculated in gen2000:

$$\sigma_{g,adj} = \sigma_g \frac{E_{g+1} - E_g}{E_{g+1} - E_{thresh}}.$$

For the example reaction, this value is  $9.946 \times 10^{-3}$  b.

Facilities are available for client code developers to multiply tabulated cross sections as they see fit to model non-analog physics.

## Projectile Multiplicities

Projectile multiplicities specify the number of projectiles of each type emerging from a reaction. Usually the multiplicity is independent of incident energy. In some cases, particularly for fission, the multiplicity is tabulated as a function of energy of the incident particle. Facilities are available for client code developers to apply a factor to projectile multiplicities in order to increase or reduce projectile production.

## Projectile Characteristic Distributions

The characteristics of outgoing projectiles are tabulated in the APM libraries in the form of equal probable distributions in either energy, angle (cosine), both energy and angle, or sometimes coupled angle-energy. The kinematics type, described below, of a particular reaction specifies which distributions are present. Typically 32 equal probable bins are used, although for some distributions, for example fission neutron spectra, more bins are used. Although this method of describing a distribution occasionally smears out some detail, it has the advantage of resolving important, high-probability portions of a distribution. It is also very computationally efficient.

Distributions are tabulated at the evaluators' choice of discrete incident energies. Linear interpolation between tabulated distributions is used to find outgoing projectile characteristics at the incident particle energy.

## Available Isotopes

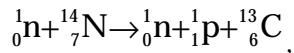
The neutron library, `mcf1.pdb`, currently contains data for 108 target isotopes and the composite isotope representing long- and short-lived fission fragments (99120 and 99125). The table “Available Targets (neutron library)” identifies these isotopes by their ZA-number (sometimes known as clyde number). An isotope with a ZA-number ending with 000 (i.e., atomic number of 000) indicates a naturally occurring element with a combination of isotopes.

The charged particle libraries, `mcf2.pdb` through `mcf6.pdb`, contain only 14 target isotopes as shown in the table “Available Targets (charged particle libraries).”

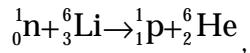
## Depletion/Accretion Data

As particles traverse material, not only are their characteristics altered but they also alter the characteristics of the background medium. The target isotope is *depleted* from the background material and the resulting products are *accreted* to the background.

The philosophy governing the generation of these libraries is “maintain a closed system” meaning that any isotope created by a reaction described in the library must itself be contained in the library. For example, the (n, n'p) reaction off  $^{14}\text{N}$ ,



creates  $^{13}\text{C}$  which is present in the library. Problems arise with reactions such as (n,p) off  $^6\text{Li}$ ,



which produces  $^6\text{He}$ , an isotope missing from the library. The library evaluators have attempted to solve this problem by substituting a listed isotope with similar A and Z, in this case  $^4\text{He}$  for  $^6\text{He}$ . This unfortunately introduces a mass error of 2 amu. Other examples are  $^8\text{Li} \rightarrow ^7\text{Li}$ ,  $^9\text{Li} \rightarrow ^7\text{Li}$ ,  $^{10}\text{Be} \rightarrow ^9\text{Be}$ ,  $^{11}\text{Be} \rightarrow ^9\text{Be}$ , etc.

**MCAPM** provides a routine for the client code to keep track of the resulting mass and charge imbalances (see “Client Code Programmer’s Manual”).

## Kinematics Physics

All reactions have been categorized into seven kinematics types, which describe the methods used to determine the characteristics of a reaction’s exiting particles. The mapping from reaction to kinematic type is another data-driven feature of the **MCAPM** package.

### Kinematics Type 1: Relativistic Two-Body

After sampling from a library center-of-mass frame angle distribution for the first exiting particle, kinematics type 1 determines the energy of the first particle and the energy and angle of the second particle from conservation of momentum and energy. It then transforms these characteristics back to the laboratory frame. If either of the outgoing particles are products rather than projectiles, its characteristics are calculated and then discarded. All elastic scattering reactions are of this type as are other reactions such as  $^7\text{Be}(\text{n},\text{p})^7\text{Li}$  and  $^{14}\text{N}(\text{n},\alpha)^{11}\text{B}$  and inelastic scattering off a level.

### Kinematics Type 2: Energy/Angle Uncorrelated

Kinematics type 2 simply samples separately from lab frame angle distributions and energy distributions for each projectile. Correlation errors abound: the angle for one particle is uncorrelated to its energy, the characteristics of one particle are uncorrelated to the characteristics of

another emerging from the same collision. Energy and momentum are only conserved on an average, statistical sense, not for each reaction. Most reactions are of type 2 including inelastic scattering off the continuum, fission, and most ( $n, 2n$ ) reactions.

### **Kinematics Type 3: Energy/Angle Correlated**

Kinematics type 3 is currently used for only one reaction:  ${}^9\text{Be}(n,2n){}^2\alpha$ . Lab frame angle distributions are sampled for each particle. Then, a lab frame energy distributions dependent on the sample cosine is sampled. The data set necessary to describe this reaction is quite voluminous because separate energy distributions exist for different exiting cosines.

### **Kinematics Type 4: Decay**

Kinematics type 4 treats the initial portion of the reaction like relativistic two-body reaction. Consider, for example,  ${}^6\text{Li}(n,nd)\alpha$ . Type 4 samples the center-of-mass angle of the exiting neutron from a library angle distribution. The neutron energy and the outgoing  ${}^6\text{Li}$  cluster's energy and angle are found in the center-of-mass frame via conservation of momentum and energy and are transformed back to the lab frame. In this model, the cluster decays immediately into its constituent parts, in this case  $d$  and  $\alpha$ . The cluster's characteristics are discarded because type 4 samples from uncorrelated lab angle and energy distributions for the remaining projectiles. In addition to  ${}^6\text{Li}(n,nd)\alpha$ , other reactions of this type include  ${}^7\text{Li}(n,n')\gamma{}^7\text{Li}$ ,  ${}^7\text{Be}(d,p\alpha)\alpha$ , and  ${}^{12}\text{C}(n,n'{}^2\alpha)\alpha$ .

### **Kinematics Type 5: Cluster**

In a manner similar to the decay method, kinematics type 5 calculates relativistic two-body kinematics for the first exiting particle and the remaining cluster. It retains the cluster angle and energy, assigning the angle and corresponding speed to the constituents of the cluster. Only one reaction,  $d(n,2n)p$ , has kinematics type 5. The first particle emerging is a neutron; the cluster is a deuteron that breaks up into a neutron and proton.

### **Kinematics Type 6: Coherent Scattering**

The exiting angular distribution of gammas experiencing coherent scattering is modeled by the Thomson cross section modified by the square of the coherent form factor:

$$\sigma_{coh}(E, \mu) = \frac{3T}{8}(1 + \mu^2)FF^2(m),$$

where

$$m = \frac{\sin(\theta/2)}{\lambda}$$

$$T = \frac{8\pi}{3} \left( \frac{e^2}{m_e c^2} \right)^2 \approx 0.665 b .$$

The form factor increases the forward peaking of the angular distribution for high-energy gammas.

The simple Thomson distribution

$$(1 + \mu^2)$$

is coded into `bang2000` and the form factors tabulated as a logarithmic interpolated function are read from the APM gamma data file. The exit angle is sampled by first sampling from the square of the coherent form factor via table lookup. The rejection method is used to apply the Thomson angular dependence to the sampled angle.

### Kinematics Type 7: Incoherent Scattering

The Klein-Nishina cross section multiplied by the incoherent form (or scattering) factor describes the angular distribution of gammas exiting an incoherent scattering event:

$$\sigma_{inc}(E, \mu) = \frac{3T}{8} \frac{1 + \mu^2 + \frac{\kappa^2(1 - \mu)^2}{1 + \kappa(1 - \mu)}}{[1 + \kappa(1 - \mu)]^2} SF(m) ,$$

where the incident energy is included in the following term

$$\kappa = \frac{E}{0.511} .$$

The effect of the form factor is to decrease the extreme forward peaking of the Klein-Nishina angular distribution at high energies.

The incoherent form factor is tabulated in the APM gamma data file as a logarithmic interpolated function. The exiting characteristics of the scattered gamma are found by sampling the exiting energy and angle from the Klein-Nishina distribution using Kahn's rejection method for gammas with incident energies below 1.5 MeV and Koblinger's direct method for higher energy gammas. The subroutine that performs this sampling was lifted directly from MCNP. The exiting angle (and corresponding energy) is then rejected or accepted by sampling an angle from the incoherent form factor with the rejection method.

## Definitions

Table 1. Projectile Particles and APM Library File

| particle      | y-number | ZA-number | file name |
|---------------|----------|-----------|-----------|
| neutron       | 1        | 1         | mcf1.pdb  |
| proton        | 2        | 1001      | mcf2.pdb  |
| deuteron      | 3        | 1002      | mcf3.pdb  |
| triton        | 4        | 1003      | mcf4.pdb  |
| $^3\text{He}$ | 5        | 2003      | mcf5.pdb  |
| alpha         | 6        | 2004      | mcf6.pdb  |
| gamma         | 7        | 0         | mcf7.pdb  |

Table 2. Energy Group Boundaries (neutron library)

| Group | Lower Boundary (MeV) |
|-------|----------------------|-------|----------------------|-------|----------------------|-------|----------------------|
|       | 1.3068e-09           | 51    | 6.0968e-05           | 101   | 5.7628e-03           | 151   | 7.5479e+00           |
| 2     | 5.2271e-09           | 52    | 6.3247e-05           | 102   | 7.5270e-03           | 152   | 7.9096e+00           |
| 3     | 2.0908e-08           | 53    | 6.5568e-05           | 103   | 1.0245e-02           | 153   | 8.3215e+00           |
| 4     | 3.2669e-08           | 54    | 6.6744e-05           | 104   | 1.5106e-02           | 154   | 8.7867e+00           |
| 5     | 4.7044e-08           | 55    | 7.0335e-05           | 105   | 2.0908e-02           | 155   | 9.1767e+00           |
| 6     | 8.3215e-08           | 56    | 7.1553e-05           | 106   | 2.6462e-02           | 156   | 9.6648e+00           |
| 7     | 1.3068e-07           | 57    | 7.5270e-05           | 107   | 3.2669e-02           | 157   | 1.0120e+01           |
| 8     | 1.8817e-07           | 58    | 7.7800e-05           | 108   | 3.9530e-02           | 158   | 1.0585e+01           |
| 9     | 2.5613e-07           | 59    | 7.9080e-05           | 109   | 4.7044e-02           | 159   | 1.1012e+01           |
| 10    | 3.3453e-07           | 60    | 8.1673e-05           | 110   | 5.7615e-02           | 160   | 1.1547e+01           |
| 11    | 4.2339e-07           | 61    | 8.4307e-05           | 111   | 7.0020e-02           | 161   | 1.1993e+01           |
| 12    | 5.1230e-07           | 62    | 8.8337e-05           | 112   | 8.3215e-02           | 162   | 1.2499e+01           |
| 13    | 7.5270e-07           | 63    | 9.1076e-05           | 113   | 9.8909e-02           | 163   | 1.3068e+01           |
| 14    | 1.1761e-06           | 64    | 9.3857e-05           | 114   | 1.3068e-01           | 164   | 1.3542e+01           |
| 15    | 1.5106e-06           | 65    | 9.6680e-05           | 115   | 1.8195e-01           | 165   | 1.3863e+01           |
| 16    | 2.0908e-06           | 66    | 9.8107e-05           | 116   | 2.0746e-01           | 166   | 1.4134e+01           |
| 17    | 2.7411e-06           | 67    | 1.0245e-04           | 117   | 2.4170e-01           | 167   | 1.4407e+01           |
| 18    | 3.5335e-06           | 68    | 1.0990e-04           | 118   | 2.7097e-01           | 168   | 1.4683e+01           |
| 19    | 4.7044e-06           | 69    | 1.1761e-04           | 119   | 2.9402e-01           | 169   | 1.5186e+01           |
| 20    | 5.6578e-06           | 70    | 1.2558e-04           | 120   | 3.3453e-01           | 170   | 1.5754e+01           |
| 21    | 6.7367e-06           | 71    | 1.3381e-04           | 121   | 3.7765e-01           | 171   | 1.6334e+01           |
| 22    | 8.3215e-06           | 72    | 1.4059e-04           | 122   | 4.2339e-01           | 172   | 1.6923e+01           |
| 23    | 9.6199e-06           | 73    | 1.5106e-04           | 123   | 5.1230e-01           | 173   | 1.7523e+01           |
| 24    | 1.1012e-05           | 74    | 1.6008e-04           | 124   | 6.3247e-01           | 174   | 1.8134e+01           |
| 25    | 1.3068e-05           | 75    | 1.6936e-04           | 125   | 7.5270e-01           | 175   | 1.8755e+01           |
| 26    | 1.4683e-05           | 76    | 1.7890e-04           | 126   | 8.8337e-01           |       | 2.0000e+01           |
| 27    | 1.5812e-05           | 77    | 1.8870e-04           | 127   | 1.0245e+00           |       |                      |
| 28    | 1.7584e-05           | 78    | 1.9876e-04           | 128   | 1.1761e+00           |       |                      |
| 29    | 1.8817e-05           | 79    | 2.0908e-04           | 129   | 1.3381e+00           |       |                      |
| 30    | 2.0746e-05           | 80    | 2.7411e-04           | 130   | 1.5106e+00           |       |                      |
| 31    | 2.2769e-05           | 81    | 3.2669e-04           | 131   | 1.6936e+00           |       |                      |
| 32    | 2.4170e-05           | 82    | 3.8105e-04           | 132   | 1.8870e+00           |       |                      |
| 33    | 2.5613e-05           | 83    | 4.7044e-04           | 133   | 2.0908e+00           |       |                      |
| 34    | 2.7097e-05           | 84    | 4.9908e-04           | 134   | 2.3051e+00           |       |                      |
| 35    | 2.8623e-05           | 85    | 5.6578e-04           | 135   | 2.5299e+00           |       |                      |
| 36    | 2.9402e-05           | 86    | 6.0425e-04           | 136   | 2.7411e+00           |       |                      |
| 37    | 3.0991e-05           | 87    | 6.3666e-04           | 137   | 3.0108e+00           |       |                      |
| 38    | 3.3453e-05           | 88    | 7.1558e-04           | 138   | 3.2669e+00           |       |                      |
| 39    | 3.6009e-05           | 89    | 8.3215e-04           | 139   | 3.5335e+00           |       |                      |
| 40    | 3.8659e-05           | 90    | 9.1767e-04           | 140   | 3.8105e+00           |       |                      |
| 41    | 4.0478e-05           | 91    | 1.0585e-03           | 141   | 4.0688e+00           |       |                      |
| 42    | 4.2339e-05           | 92    | 1.3068e-03           | 142   | 4.3960e+00           |       |                      |
| 43    | 4.3285e-05           | 93    | 1.5812e-03           | 143   | 4.7044e+00           |       |                      |
| 44    | 4.6186e-05           | 94    | 1.8817e-03           | 144   | 4.9908e+00           |       |                      |
| 45    | 4.7174e-05           | 95    | 2.2084e-03           | 145   | 5.3525e+00           |       |                      |
| 46    | 4.9181e-05           | 96    | 2.5613e-03           | 146   | 5.6578e+00           |       |                      |
| 47    | 5.1230e-05           | 97    | 2.9402e-03           | 147   | 6.0425e+00           |       |                      |
| 48    | 5.3321e-05           | 98    | 3.3453e-03           | 148   | 6.3666e+00           |       |                      |
| 49    | 5.6536e-05           | 99    | 3.7765e-03           | 149   | 6.7367e+00           |       |                      |
| 50    | 5.7628e-05           | 100   | 4.2339e-03           | 150   | 7.1558e+00           |       |                      |

Table 3. Energy Group Boundaries (charged particle libraries)

| Group | Lower Boundary (MeV) |
|-------|----------------------|-------|----------------------|-------|----------------------|-------|----------------------|
| 1     | 1.0000e-10           | 21    | 2.3826e-01           | 41    | 2.3223e+00           | 61    | 1.5103e+01           |
| 2     | 2.0000e-02           | 22    | 2.7144e-01           | 42    | 2.5502e+00           | 62    | 1.6585e+01           |
| 3     | 2.2786e-02           | 23    | 3.0925e-01           | 43    | 2.8005e+00           | 63    | 1.8213e+01           |
| 4     | 2.5959e-02           | 24    | 3.5232e-01           | 44    | 3.0753e+00           |       | 2.0000e+01           |
| 5     | 2.9575e-02           | 25    | 4.0140e-01           | 45    | 3.3771e+00           |       |                      |
| 6     | 3.3694e-02           | 26    | 4.5730e-01           | 46    | 3.7085e+00           |       |                      |
| 7     | 3.8388e-02           | 27    | 5.2100e-01           | 47    | 4.0725e+00           |       |                      |
| 8     | 4.3735e-02           | 28    | 5.9357e-01           | 48    | 4.4721e+00           |       |                      |
| 9     | 4.9826e-02           | 29    | 6.7624e-01           | 49    | 4.9110e+00           |       |                      |
| 10    | 5.6766e-02           | 30    | 7.7043e-01           | 50    | 5.3930e+00           |       |                      |
| 11    | 6.4673e-02           | 31    | 8.7774e-01           | 51    | 5.9222e+00           |       |                      |
| 12    | 7.3681e-02           | 32    | 1.0000e+00           | 52    | 6.5034e+00           |       |                      |
| 13    | 8.3943e-02           | 33    | 1.0981e+00           | 53    | 7.1417e+00           |       |                      |
| 14    | 9.5635e-02           | 34    | 1.2059e+00           | 54    | 7.8426e+00           |       |                      |
| 15    | 1.0896e-01           | 35    | 1.3242e+00           | 55    | 8.6122e+00           |       |                      |
| 16    | 1.2413e-01           | 36    | 1.4542e+00           | 56    | 9.4574e+00           |       |                      |
| 17    | 1.4142e-01           | 37    | 1.5969e+00           | 57    | 1.0386e+01           |       |                      |
| 18    | 1.6112e-01           | 38    | 1.7537e+00           | 58    | 1.1405e+01           |       |                      |
| 19    | 1.8356e-01           | 39    | 1.9258e+00           | 59    | 1.2524e+01           |       |                      |
| 20    | 2.0913e-01           | 40    | 2.1147e+00           | 60    | 1.3753e+01           |       |                      |

Table 4. Energy Points (gamma library)

| Index | Energy (MeV) |
|-------|--------------|-------|--------------|-------|--------------|-------|--------------|
| 1     | 1.0000E-04   | 51    | 3.7000E-03   | 101   | 2.3120E-02   | 151   | 9.0910E-01   |
| 2     | 1.0890E-04   | 52    | 3.8280E-03   | 102   | 2.5530E-02   | 152   | 1.0000E+00   |
| 3     | 1.1030E-04   | 53    | 4.0300E-03   | 103   | 2.9180E-02   | 153   | 1.1250E+00   |
| 4     | 1.4660E-04   | 54    | 4.1500E-03   | 104   | 3.3160E-02   | 154   | 1.2660E+00   |
| 5     | 1.5000E-04   | 55    | 4.3000E-03   | 105   | 3.7500E-02   | 155   | 1.4240E+00   |
| 6     | 2.0000E-04   | 56    | 4.4960E-03   | 106   | 4.2000E-02   | 156   | 1.6020E+00   |
| 7     | 2.1580E-04   | 57    | 4.8100E-03   | 107   | 4.7360E-02   | 157   | 1.8020E+00   |
| 8     | 2.1720E-04   | 58    | 5.1630E-03   | 108   | 4.8510E-02   | 158   | 2.0270E+00   |
| 9     | 2.5010E-04   | 59    | 5.4670E-03   | 109   | 5.0230E-02   | 159   | 2.2810E+00   |
| 10    | 2.5150E-04   | 60    | 5.5460E-03   | 110   | 5.5000E-02   | 160   | 2.5660E+00   |
| 11    | 3.0000E-04   | 61    | 5.9340E-03   | 111   | 6.1000E-02   | 161   | 2.8870E+00   |
| 12    | 3.2440E-04   | 62    | 5.9800E-03   | 112   | 6.7450E-02   | 162   | 3.2470E+00   |
| 13    | 3.8470E-04   | 63    | 6.2500E-03   | 113   | 6.7460E-02   | 163   | 3.6530E+00   |
| 14    | 3.8810E-04   | 64    | 6.5430E-03   | 114   | 6.9510E-02   | 164   | 4.1100E+00   |
| 15    | 4.0000E-04   | 65    | 6.7500E-03   | 115   | 7.3500E-02   | 165   | 4.6240E+00   |
| 16    | 4.3740E-04   | 66    | 6.9800E-03   | 116   | 7.8350E-02   | 166   | 5.2020E+00   |
| 17    | 4.3900E-04   | 67    | 7.1130E-03   | 117   | 8.0670E-02   | 167   | 5.8520E+00   |
| 18    | 5.0000E-04   | 68    | 7.2510E-03   | 118   | 8.5520E-02   | 168   | 6.5830E+00   |
| 19    | 5.5000E-04   | 69    | 7.7130E-03   | 119   | 8.7950E-02   | 169   | 7.4060E+00   |
| 20    | 5.9430E-04   | 70    | 7.9420E-03   | 120   | 9.0540E-02   | 170   | 8.3320E+00   |
| 21    | 5.9470E-04   | 71    | 8.1800E-03   | 121   | 9.2300E-02   | 171   | 9.3730E+00   |
| 22    | 6.0000E-04   | 72    | 8.3830E-03   | 122   | 9.6000E-02   | 172   | 1.0550E+01   |
| 23    | 6.8360E-04   | 73    | 8.6000E-03   | 123   | 1.0000E-01   | 173   | 1.1860E+01   |
| 24    | 6.8500E-04   | 74    | 8.9820E-03   | 124   | 1.0500E-01   | 174   | 1.3350E+01   |
| 25    | 7.0000E-04   | 75    | 9.2000E-03   | 125   | 1.0970E-01   | 175   | 2.0000E+01   |
| 26    | 7.5410E-04   | 76    | 9.4000E-03   | 126   | 1.0980E-01   | 176   | 3.0000E+01   |
| 27    | 7.5550E-04   | 77    | 9.5600E-03   | 127   | 1.1490E-01   |       |              |
| 28    | 8.0000E-04   | 78    | 9.6600E-03   | 128   | 1.1500E-01   |       |              |
| 29    | 8.1970E-04   | 79    | 9.8000E-03   | 129   | 1.2110E-01   |       |              |
| 30    | 8.2110E-04   | 80    | 1.0010E-02   | 130   | 1.2120E-01   |       |              |
| 31    | 9.0000E-04   | 81    | 1.0390E-02   | 131   | 1.3510E-01   |       |              |
| 32    | 1.0000E-03   | 82    | 1.0750E-02   | 132   | 1.4860E-01   |       |              |
| 33    | 1.0800E-03   | 83    | 1.1140E-02   | 133   | 1.6350E-01   |       |              |
| 34    | 1.1400E-03   | 84    | 1.1920E-02   | 134   | 1.7990E-01   |       |              |
| 35    | 1.3600E-03   | 85    | 1.2650E-02   | 135   | 1.9780E-01   |       |              |
| 36    | 1.5590E-03   | 86    | 1.3030E-02   | 136   | 2.1760E-01   |       |              |
| 37    | 1.7000E-03   | 87    | 1.3860E-02   | 137   | 2.3940E-01   |       |              |
| 38    | 1.8380E-03   | 88    | 1.4700E-02   | 138   | 2.6330E-01   |       |              |
| 39    | 1.9100E-03   | 89    | 1.5200E-02   | 139   | 2.8970E-01   |       |              |
| 40    | 2.1420E-03   | 90    | 1.5710E-02   | 140   | 3.1860E-01   |       |              |
| 41    | 2.2400E-03   | 91    | 1.6390E-02   | 141   | 3.5050E-01   |       |              |
| 42    | 2.4100E-03   | 92    | 1.7000E-02   | 142   | 3.8550E-01   |       |              |
| 43    | 2.4700E-03   | 93    | 1.7160E-02   | 143   | 4.2410E-01   |       |              |
| 44    | 2.6300E-03   | 94    | 1.8060E-02   | 144   | 4.6650E-01   |       |              |
| 45    | 2.7500E-03   | 95    | 1.8980E-02   | 145   | 5.1320E-01   |       |              |
| 46    | 2.8200E-03   | 96    | 1.9670E-02   | 146   | 5.6450E-01   |       |              |
| 47    | 2.9600E-03   | 97    | 2.0100E-02   | 147   | 6.2090E-01   |       |              |
| 48    | 3.1800E-03   | 98    | 2.0950E-02   | 148   | 6.8300E-01   |       |              |
| 49    | 3.5470E-03   | 99    | 2.1750E-02   | 149   | 7.5130E-01   |       |              |
| 50    | 3.6060E-03   | 100   | 2.2250E-02   | 150   | 8.2640E-01   |       |              |

Table 5. Available Targets (neutron library)

| ZA-number | Isotope    | ZA-number | Isotope    | ZA-number | Isotope          |
|-----------|------------|-----------|------------|-----------|------------------|
| 1         | Neutron    | 29000     | Cu natural | 92233     | U 233            |
| 1001      | H 1        | 30000     | Zn natural | 92234     | U 234            |
| 1002      | H 2        | 31000     | Ga natural | 92235     | U 235            |
| 1003      | H 3        | 33074     | As 74      | 92236     | U 236            |
| 2003      | He 3       | 33075     | As 75      | 92237     | U 237            |
| 2004      | He 4       | 39088     | Y 88       | 92238     | U 238            |
| 3006      | Li 6       | 39089     | Y 89       | 92239     | U 239            |
| 3007      | Li 7       | 40000     | Zr natural | 92240     | U 240            |
| 4007      | Be 7       | 41093     | Nb 93      | 93235     | Np 235           |
| 4009      | Be 9       | 42000     | Mo natural | 93236     | Np 236           |
| 5010      | B 10       | 47107     | Ag 107     | 93237     | Np 237           |
| 5011      | B 11       | 47109     | Ag 109     | 93238     | Np 238           |
| 6012      | C 12       | 48000     | Cd natural | 94237     | Pu 237           |
| 6013      | C 13       | 49000     | In natural | 94238     | Pu 238           |
| 7014      | N 14       | 50000     | Sn natural | 94239     | Pu 239           |
| 7015      | N 15       | 51000     | Sb natural | 94240     | Pu 240           |
| 8016      | O 16       | 53127     | I 127      | 94241     | Pu 241           |
| 9019      | F 19       | 54000     | Xe natural | 94242     | Pu 242           |
| 10020     | Ne 20      | 54134     | Xe 134     | 94243     | Pu 243           |
| 11023     | Na 23      | 56138     | Ba 138     | 95241     | Am 241           |
| 12000     | Mg natural | 63000     | Eu natural | 95242     | Am 242           |
| 13027     | Al 27      | 64000     | Gd natural | 95243     | Am 243           |
| 14000     | Si natural | 67165     | Ho 165     | 96242     | Cm 242           |
| 15031     | P 31       | 72000     | Hf natural | 96243     | Cm 243           |
| 16032     | S 32       | 73181     | Ta 181     | 96244     | Cm 244           |
| 17000     | Cl natural | 74000     | W natural  | 96245     | Cm 245           |
| 18000     | Ar natural | 75185     | Re 185     | 96246     | Cm 246           |
| 19000     | K natural  | 75187     | Re 187     | 96247     | Cm 247           |
| 20000     | Ca natural | 78000     | Pt natural | 96248     | Cm 248           |
| 22000     | Ti natural | 79197     | Au 197     | 97249     | Bk 249           |
| 23051     | V 51       | 80000     | Hg natural | 98249     | Cf 249           |
| 24000     | Cr natural | 82000     | Pb natural | 98250     | Cf 250           |
| 25055     | Mn 55      | 83209     | Bi 209     | 98251     | Cf 251           |
| 26000     | Fe natural | 90231     | Th 231     | 98252     | Cf 252           |
| 27059     | Co 59      | 90232     | Th 232     | 99120     | fission products |
| 28000     | Ni natural | 90233     | Th 233     | 99125     | fission products |
| 28058     | Ni 58      | 91233     | Pa 233     |           |                  |

Table 6. Available Targets (charged particle libraries)

| ZA-number | Isotope | ZA-number | Isotope | ZA-number | Isotope |
|-----------|---------|-----------|---------|-----------|---------|
| 1001      | H 1     | 3006      | Li 6    | 5011      | B 11    |
| 1002      | H 2     | 3007      | Li 7    | 6012      | C 12    |
| 1003      | H 3     | 4007      | Be 7    | 7014      | N 14    |
| 2003      | He 3    | 4009      | Be 9    | 8016      | O 16    |
| 2004      | He 4    | 5010      | B 10    |           |         |

Table 7. Available Targets (gamma library)

| ZA-number | Element | ZA-number | Element | ZA-number | Element          |
|-----------|---------|-----------|---------|-----------|------------------|
| 1000      | H       | 36000     | Kr      | 71000     | Lu               |
| 2000      | He      | 37000     | Rb      | 72000     | Hf               |
| 3000      | Li      | 38000     | Sr      | 73000     | Ta               |
| 4000      | Be      | 39000     | Y       | 74000     | W                |
| 5000      | B       | 40000     | Ar      | 75000     | Re               |
| 6000      | C       | 41000     | Nb      | 76000     | Os               |
| 7000      | N       | 42000     | Mo      | 77000     | Ir               |
| 8000      | O       | 43000     | Tc      | 78000     | Pt               |
| 9000      | F       | 44000     | Ru      | 79000     | Au               |
| 10000     | Ne      | 45000     | Rh      | 80000     | Hg               |
| 11000     | Na      | 46000     | Pd      | 81000     | Tl               |
| 12000     | Mg      | 47000     | Ag      | 82000     | Pb               |
| 13000     | Al      | 48000     | Cd      | 83000     | Bi               |
| 14000     | Si      | 49000     | In      | 84000     | Po               |
| 15000     | P       | 50000     | Sn      | 85000     | At               |
| 16000     | S       | 51000     | Sb      | 86000     | Rn               |
| 17000     | Cl      | 52000     | Te      | 87000     | Fr               |
| 18000     | Ar      | 53000     | I       | 88000     | Ra               |
| 19000     | K       | 54000     | Xe      | 89000     | Ac               |
| 20000     | Ca      | 55000     | Cs      | 90000     | Th               |
| 21000     | Sc      | 56000     | Ba      | 91000     | Pa               |
| 22000     | Ti      | 57000     | La      | 92000     | U                |
| 23000     | V       | 58000     | Ce      | 93000     | Np               |
| 24000     | Cr      | 59000     | Pr      | 94000     | Pu               |
| 25000     | Mn      | 60000     | Nd      | 95000     | Am               |
| 26000     | Fe      | 61000     | Pm      | 96000     | Cm               |
| 27000     | Co      | 62000     | Sm      | 97000     | Bk               |
| 28000     | Ni      | 63000     | Eu      | 98000     | Cf               |
| 29000     | Cu      | 64000     | Gd      | 99000     | Es               |
| 30000     | Zn      | 65000     | Tb      | 99120     | fission products |
| 31000     | Ga      | 66000     | Dy      | 100000    | Fm               |
| 32000     | Ge      | 67000     | Ho      |           |                  |
| 33000     | As      | 68000     | Er      |           |                  |
| 34000     | Se      | 69000     | Tm      |           |                  |
| 35000     | Br      | 70000     | Yb      |           |                  |

Table 8. Reaction IDs (C-numbers)

| C  | reaction description               | C  | reaction description            |
|----|------------------------------------|----|---------------------------------|
| 8  | large angle coulomb scattering     | 46 | y, $\gamma$                     |
| 9  | nuclear scattering and interaction | 47 | y,d $\alpha$                    |
| 10 | Elastic                            | 48 | y,p $\alpha$                    |
| 11 | y,n'                               | 49 | y,2p $\alpha$                   |
| 12 | y,2n                               | 50 | y,Xp                            |
| 13 | y,3n                               | 51 | y,Xd                            |
| 14 | y,4n                               | 52 | y,Xt                            |
| 15 | y,f                                | 53 | y,X <sup>3</sup> He             |
| 16 | y,3np                              | 54 | y,X $\alpha$                    |
| 17 | y,n2p                              | 55 | y,X $\gamma$                    |
| 18 | y,2p                               | 56 | y,Xn                            |
| 19 | y,pd                               | 57 | y,Xe                            |
| 20 | y,n'p                              |    |                                 |
| 21 | y,pn'                              | 65 | Activation                      |
| 22 | y,n'd                              | 66 | Yields                          |
| 23 | y,n'd $\alpha$                     |    |                                 |
| 24 | y,n't                              | 70 | Total photon                    |
| 25 | y,n <sup>3</sup> He                | 71 | Coherent scattering             |
| 26 | y,n' $\alpha$                      | 72 | Incoherent scattering           |
| 27 | y,n'2 $\alpha$                     | 73 | Photoelectric                   |
| 28 | y,n't $\alpha$                     | 74 | Pair production (nuclear)       |
| 29 | y,2np                              | 75 | Pair production (electronic)    |
| 30 | y,gn $\alpha$                      | 78 | Internal conversion             |
| 31 | y,2np $\alpha$                     | 81 | Ionization                      |
| 32 | y,2nd                              | 82 | Bremsstrahlung                  |
| 33 | y,2n' $\alpha$                     | 83 | Excitation                      |
| 34 | y,np $\alpha$                      | 84 | Collision                       |
| 35 | y,dn'                              | 91 | Atomic subshell parameters      |
| 37 | y,2 $\alpha$                       | 92 | Subshell transition probability |
| 38 | y, <sup>3</sup> He, $\alpha$       | 93 | Whole atom parameters           |
| 39 | y,tp                               |    |                                 |
| 40 | y,p                                |    |                                 |
| 41 | y,d                                |    |                                 |
| 42 | y,t                                |    |                                 |
| 43 | y,t $\alpha$                       |    |                                 |
| 44 | y, <sup>3</sup> He                 |    |                                 |
| 45 | y, $\alpha$                        |    |                                 |

Table 9. Code Units

|        |                                  |
|--------|----------------------------------|
| Time   | Shake (1.0×10 <sup>-8</sup> sec) |
| length | cm                               |
| speed  | cm/shake                         |
| energy | MeV                              |

## Client Code Programmer's Manual

This section of *MCAPM Generator and Collision Routine Documentation* should provide client code programmers or developers the information necessary to integrate gen2000 and bang2000 into their Monte Carlo transport package. I assume the reader is familiar with the previous section of the report, the “User’s Manual.” Following a brief overview of the usage of the package and descriptions of routines available to access data from the internal data library generated by gen2000, topics covered are roughly divided into physics issues and computer science issues. Physics topics are non-analog capability, particle sourcing options, group collapse, and the origin of energy deposition. Computer science topics are treatments of secondary particle index, routines that must be supplied by client code programmers, machine portability (i.e., deviations from standard C), and mechanics of lib and tar files. This section ends with excerpts from routines with which client code developers must interact. These excerpts provide interface information and brief descriptions of the routines. We conclude with a listing of the `mcapm.h` header file.

### Usage Overview

The generator, `gen2000`, must be called for each particle library desired before any collision physics can be performed by `bang2000`. The client code provides `gen2000` with the y-number (1-7) of the incident particle and a list of ZA-numbers of the isotopes desired. This list will be sorted by `gen2000` to include only isotopes present in the library. This can be avoided by first previewing the library with the routine `PreViewC` to determine the library contents. `gen2000` returns a number of parameters, but the critical parameter is a pointer to the internal library created for the specified incident particle.

When `bang2000` is called to perform collision physics, the pointer corresponding to the appropriate incident particle must be passed together with data describing the incident particle (group index and energy) and the composition of the background material (isotopic number densities). `bang2000` will use the library pointed to by the pointer given by `gen2000` to determine what reaction occurred, the exiting projectiles’ characteristics (number, type, energy, speed, and cosine), and accretion products (number and ZA-numbers). Projectile characteristics are returned in the target frame relative to the incident particle’s direction. It is the responsibility of the client code to introduce a random azimuthal angle, rotate from the incident particle direction to problem coordinates, and account for any target motion.

The collision physics performed by `bang2000` consists of three parts:

- 1) sampling the target isotope (`SmpIso`),
- 2) sampling the reaction (`SmpRct`),
- 3) sampling the source of resulting particles (`SmpSrc`).

Furthermore there are the 2 special sampling routines

- 1) sampling the isotope and the reaction (`SmpIsRc`), and
- 2) sampling the multi-band reaction (`SmpRctMB`).

If desired, the client code can call the subroutines noted in parentheses to explicitly perform these functions instead of allowing `bang2000` to do so.

## Access Routines

Two types of access routines are available to developers so they can access contents of the library without needing to know its format. The first type (`PreViewC`) allows developers to preview the contents of a particle library prior to calling `gen2000`. The second type (`BalancZAC`, `Dat0AccC`, `Dat1AccC`, `DistEinSecRAccC`, `DistEinSecRLibAccC`, `EgpSecRAcSbC`, `EinSecRAccC`, `Iso2AccC`, `PrdIsoAccC`, `PrdRctAccC`, `RctGrpAccC`, `stCalc`) provides more extensive access and must be called after `gen2000`. The second type will return different data depending on input.

Table 10. Data Available from gen2000 Access Routines

| Access Routine | Data Available |  |
|----------------|----------------|--|
|                | Type           | Options  |
| PreViewC       | Misc.          | Library date<br>Isotope ZA numbers<br>Temperatures<br>Number of energy groups  |
| BalancZAC      | Scalars        | Balance of Z,A   |
| Dat0AccC       | Scalars        | y-number of incident particle<br>ZA-number of incident particle<br>Atomic weight of incident particle<br>Excitation energy of incident particle<br>Number of energy groups<br>Number of isotopes<br>Number of projectiles<br>Number of particle descriptors, y<br>Source mode control for all particles<br>Library type<br>Cross section interpolation type<br>$\bar{v}$ interpolation type<br>multiplicity interpolation type<br>energy deposition interpolation type |
| Dat1AccC       | "1-D"          | ZA number by isotope<br>Number of reactions by isotope<br>Atomic weight by isotope<br>Excitation energy by isotope<br>Temperature index by isotope<br>Energy group boundaries by energy group<br>Temperatures<br>y-number by projectile<br>atomic weight by projectile<br>mapping from y-number to projectile index<br>ZA number by projectile   |

## Data Available from gen2000 Access Routines (cont'd)

| Access Routine     | Data Available   |  |
|--------------------|--|--|
|                    | Type   | Options  |
| DistEinSecRAccC    | "1-D" or<br>"2-D"  | Equal-probable-bin arrays for outgoing<br>- energy or<br>- angle or<br>- angle/energy correlated   |
| DistEinSecRLibAccC | "1-D" or<br>"2-D"  | As 'DistEinSecRAccC' but in COM frame for<br>Kintypes 1,4,5  |
| EgpSecRAcSbC       | "1-D"  | Multiplicity by energy group   |
| EinSecRAccC        | "1-D"  | Table of incident energies for equal-probable-bin<br>arrays for outgoing<br>- energy or<br>- angle or<br>- angle/energy correlated   |
| Iso2AccC           | Reaction or<br>energy group<br>data by<br>specified<br>isotope | C-number by reaction<br>S-number by reaction<br>Mass-balance Q by reaction<br>Reaction threshold by reaction<br>Kinematics type by reaction<br>Number of projectiles by reaction<br>y-numbers of projectiles by reaction<br>number of products by reaction<br>atomic weight of "2-Body" products by reaction<br>excitation of "2-Body" products by reaction<br>total microscopic cross section by energy group<br>number of interaction reactions by energy group<br>total number of reactions by energy group<br>local energy deposition averaged over all reactions<br>(MeV-b) by energy group<br>energy production averaged over all reactions (MeV-<br>b) by energy group<br>gain (neutron production) averaged over all reactions<br>(v-1) by energy group<br>energy-independent multiplicities of projectiles by<br>reaction<br>number of coherent form factors<br>number of incoherent form factors |
| PrdIsoAccC         | Product data<br>by specified<br>isotope                        | ZA number of products  |
| PrdRctAccC         | Product data<br>by specified<br>isotope and<br>reaction        | ZA number of products<br>Atomic weight of products<br>coherent/incoherent photon scattering form factors   |

## Data Available from gen2000 Access Routines (cont'd)

| Routine    | Data Available                                      |   |
|------------|---|---|
|            | Type  | Options   |
| RctGrpAccC | energy group data by specified isotope and reaction | Microscopic reaction cross section by energy group<br>local energy deposition by energy group<br>energy production (MeV-b) by energy group<br>gain (neutron production) (v-1) by energy group |
| stCalc     |   | total macroscopic cross section for given composition   |

**Customization Routines**

A user code can modify the microscopic cross section for a specified isotope and reaction using the routines `XSecModC` and `XSecMultC`. The cross section can be changed by either multiplying it with a constant, adding a constant, or specifying an absolute value. A user-specified reaction can be eliminated from the reaction sampling list with the routine `ElimReact`, and the reaction product array is modifiable with `ZASubMisc`.

Table 11. Data accessible by MCAPM customization routines

| Routine   | Modifiable Parameters                                   |   |
|-----------|---|---|
|           | Type  | Options   |
| ElimReact | Eliminate reaction for specified isotope                |   |
| XSecModC  | Modify cross section for specified isotope and reaction | Multiply, add, or set microscopic cross section<br>May be energy group independent or Dependent   |
| XSecMultC | Modify cross section for specified isotope and reaction | Multiplies microscopic cross section (currently not available in <code>libmcapm.a</code> )  |
| ZASubMisc | Modify reaction product array                           | Replace ZA numbers in product array with indices<br>Substitute isotopes if not available in library (currently not available in <code>libmcapm.a</code> ) |

## Non-Analog Physics

**MCAPM** provides the capability to model a slightly twisted version of reality. This may be desired in order to improve Monte Carlo statistics or to augment the effect of a particular reaction. At present **MCAPM** does not utilize the concept of particle weights. If client code programmers wish to boost gamma production, for example, they must reduce gamma weights accordingly in order to preserve a “fair game”. For support of analytical test problems, **MCAPM** provides 2 routines (`PnInit`, `PnSamp`) for sampling from a user-defined angular distribution.

## Sourcing Modes-Reaction and Emission

Two sourcing modes are supported by `bang2000`: *reaction*, where the emitted particles are those that result from the sampled reaction, and *emission*, where all possible secondary particles resulting from the reaction-inducing particle and target are sampled. Secondary particles emitted through reaction samples are correlated to the sampled reaction with one exception. Data evaluators attempt to associate every particle exiting from a interaction experiment to a specific reaction. Many exiting particles, particularly in the case of gammas cannot be attributed to specific reaction. LLNL has taken the approach of assigning these “un-associated” particles to special “reaction” (or C) numbers ranging from 50-57. Gammas are assigned the number 55. When a client code chooses reaction sampling, `bang2000` first emits those particles resulting from the sampled reaction. Then, the left over C=55 gammas are sampled with the appropriate probability from the provided energy and angle distributions.

Client code must provide adequate space for sourcing routines to store outgoing particle characteristics. The exact size required is unfortunately not known until the particles are actually produced. Currently no protection is provided by `bang2000` against producing more particles than for which there is space available.

## Group Collapse

Library cross sections can be collapsed through the use of three arguments to `gen2000`. The client code developer provides `gen2000` with the number of groups desired. If this number is less than the original number of fine groups, the developer provides a mapping of fine to coarse groups and a fine group weighting function. The collapsed cross section is

$$\sigma_G = \frac{\sum_g w_g \sigma_g \Delta E_g}{\sum_g w_g \Delta E_g},$$

where `G=nCrsFin(g)`, the fine to coarse group mapping, and `wg=xFinWght(g)`, the fine group weighting function, both arguments to `gen2000`.

## Energy Deposition

The local energy deposition values available through the gen2000 access routines `Iso2AccC` for energy deposition cross-section-weighted over all reactions of an isotope) and `RctGrpAccC` (for individual isotope-reaction energy deposition) provide a means for client codes to conserve zone energies in a statistical, average sense. These values, the average energy deposited by all *products*, are the difference of the total energy available (the incident particle's energy plus the reaction's Q) and the sum of the average, expected value of the energy carried away by the *projectiles* specified by the client code. All the fundamental energy data are read from the library data file.

## Random Number Generator

The random number generator employed in **MCAPM** is user-defined and specified by (1) a generator function, and (2) its current generator state (seed). This is a useful feature in parallel computing applications, where it allows a per-processor or per-thread random number state. Alternatively, per-particle seeding enables fine-detail deterministic replays of collision histories in a parallel processor environment. A thread-safe random number package (**RNG**) for these purposes is available on **LC** at `/usr/apps/rng`. The package consists of a Linear Congruential Generator (`lcg64`), and three different Prime Modulus Congruential Generators (`pmlcg61`, `pmlcg62_16`, `pmlcg64_10`). More details on the algorithms and the calling syntax (in C or Fortran) can be found in the **RNG** documentation. A system-supplied generator (e.g. `drand48`) may also be called (from C or Fortran) by passing the system function as the generator, and a NULL pointer as the state.

## Library Access

The data libraries are stored in portable binary format (pdb) and can be accessed on a variety of platforms (IBM AIX, Compaq Unix, Linux, SGI Irix, Sun Solaris etc.) with the Portable Applications Code Toolkit (PACT). More information on the installation and features of PACT can be found at [http://www.llnl.gov/def\\_sci/pact/](http://www.llnl.gov/def_sci/pact/). On **LC**, the data libraries are installed in `/usr/apps/mcapm/data`. The current versions of the **MCAPM** library file `libmcapm.a` and include file `mcapm.h` are located at `/usr/apps/mcapm/lib`, and `/usr/apps/mcapm/include`. Newly released beta versions of **MCAPM** can be found in `/usr/apps/mcapm/betas`.

**MCAPM** is available under UNIX in a tar library `mcapm.tar`. Contained in the tar file are RCS (Revision Control System) repositories for all the source files, include files, and a single Makefile executable under IBM AIX, Compaq Unix, Linux, SGI Irix, or Sun Solaris. When invoked, the makefile compiles sources, creates an archive (ar) files named `libmcapm.a` containing all routines that need be loaded by the client code on a UNIX system, and loads the contents of `libmcapm.a` with some additional testing routines to make a test program. The makefile can be invoked with options to perform subsets of these tasks.

## Future Improvements

Some improvements planned as time permits :

- increased performance
- electron transport
- continuous cross section treatment

## gen2000/bang2000 Calling Sequences

This section provides the first few lines that define interfaces to routines that might be called by the client code and subroutines called by **MCAPM** that might be replaced by the client code developer. In a few cases, entire routines are provided.

### Library Previewer

#### PreViewC

```
void PreViewC(int *iyInc,
              int *libdaty, int *nZAInc, int *MatNum, int **nZA,
              int *NumTmp, float **Tmp, int *maxEG, int *ierr,
              int *ioLib)
/*
Description
    preview library, which is opened and closed in this routine
Input
    *iyInc      - 'yi' of incident particle
    *ioLib      - not used
Output
    *libdaty   - library date
    *nZAInc    - 'ZA' of incident particle
    *MatNum    - number of targets
    *nZA       - pointer to 'nZA', target ZA numbers
    *NumTmp    - number of temperatures
    *Tmp        - pointer to 'Tmp', temperatures in library
    *maxEG     - number of energy groups
    *ierr       - error message
                0 for no error
                9 file for 'iyInc' not found
*/
```

### Generator

#### gen2000

```
void gen2000(int *iGiso, int *nGiso, int *iTReqd, int *iyProjIn,
              int *nProjU,
              int *iyIncP, SourceMode *jSrcCtr,
              int *maxEG, int *nCrsFin, float *xFinWght,
              int *libdaty, int *nZAInc, int *NumTmp, int *nYno,
              LibraryType *LibTyp, InterpType *ITSig, InterpType *ITEDep,
              int *jGiso, int *nZAGiso, int *nRS,
              SuperLibrary **SupLibP, int *LenSupLib,
              int *ierr,
              int *ioLib, int *itoout, int *itoerr)
/*
Input
```

```

*igiso      - array
             - desired global isotope ZA numbers
*nGiso      - pointer
             - number of desired isotopes
*itReqd     - pointer (not currently used)
*iyProjIn   - array
             - 'yo's of projectiles to track
*nProjU     - pointer
             - number of projectiles to track
*iyIncP     - pointer
             - 'yi' of incident particle
*jSrcCtr    - array
             - source control by projectile
REACTION (=1) - reaction sourcing mode:
                 outgoing particles are correlated to reaction
                 when correlation is known;
                 use emission for remaining, uncorrelated
                 sources
EMISSION (=2) - emission sourcing mode:
                 outgoing particles depend only on isotope
                 reacting, not particluar reaction
*maxEG      - number of energy groups requested by client
*nCrsFin   - collapsing map ->
              nCrsFin[nEGFin] - coarse group for fine group 'nEGFin'
*xFinWght  - fine group weighting ->
              xFinWght[nEGFin] - weighting for fine group 'nEGFin'
*ITSig      - (input and output)
              cross section interpolation type requested (passed)
cross section interpolation type used
only relevant if '*LibTyp' == FIXEDGRID
This argument allows the calling routine to override
the interpolation type specified by the library file.
To override, pass a value >= 10. gen2000 will then
subtract 10 from 'ITSig' and use the new value for
the interpolation type, as follows:


| input value | output value | y-var? | x-var? |
|-------------|--------------|--------|--------|
| 10          | 0            | lin    | lin    |
| 12          | 2            | lin    | lin    |
| 13          | ERROR        | lin    | log    |
| 14          | ERROR        | log    | lin    |
| 15          | 5            | log    | log    |


*LibTyp    - =0 take the default, group wise cross section
             = POINTWISE, take point wise cross section.
Output
*libdaty   - pointer
             - library date (passed)
*nZAInc    - pointer
             - 'ZA' of incident particle
*NumTmp    - pointer (not currently used)
*nYno      - pointer
             - number of y-numbers in library file
*ITSig     - pointer
             - cross section interpolation type used
*ITEDep    - pointer
             - energy deposition interpolation type
             only relevant if '*LibTyp' == FIXEDGRID
interpolation types


| value | y-var?       | x-var? |
|-------|--------------|--------|
| -1    | not relevant |        |
| 0     | lin          | lin    |
| 2     | lin          | lin    |
| 3     | lin          | log    |
| 4     | log          | lin    |


```

```

      5      log      log
*jGiso   - pointer
          - number of found isotopes
*nZAGiso - array
          - ZA numbers of found targets
*nRS     - array
          - number of reactions by isotope
**SupLibP - handle to library
*LenSupLib- pointer (not currently used)
*ierr     - pointer
          - error flag
          0 for no error
          1 if an isotope in 'iGiso' not found
          9 for 'iyIncP' not found
*ioLib    - pointer (not currently used)
*itoout   - pointer (not currently used)
*itoerr   - pointer (not currently used)
*/

```

## Access Routines

### BalancZAC

```

void BalancZAC(int *nZAtrg,
                int *nSecProj, int *iSecProj, int *Mult,
                int *nProd, int *nZAProd,
                int *nZAProj, int *nZAiyi, int *nZAInc,
                int *nZBal, int *nABal, int *nZABal)
/*
Description
calculate balance of protons and nucleons
Input
*nZAtrg  - ZA of target
*nSecProj - number of secondary particles that are projectiles
iSecProj - 'yo's or projectile index of secondary particles
Mult     - multiplicity of secondary particles
         (-1 if energy-dependent)
*nProd    - number of "mass balance" products
nZAProd  - ZA number of products
nZAProj  - ZA numbers of projectiles by projectile index
nZAiyi   - ZA numbers of projectiles by iyi index
*nZAInc   - ZA of incident particle
Output
*nZBal   - balance of Z number
*nABal   - balance of A number
*nZABal  - balance of ZA number
*/

```

**Dat0AccC**

```

int Dat0AccC(SuperLibrary *SupLib,
              int *iyIncP, int *nZAIncP, float *AWIncP,
              float *xLevIncP, int *maxEGx, int *jGisoP, int *nProjUP,
              int *nYnoP, int *iSrcCtrP, int *LibTypeP,
              int *ITSigP, int *ITNuP, int *ITMultP, int *ITEDepP)
/*
Description
    access scalar data
    (This subroutine must not be called prior to calling 'gen2000'.)
Input
    *SupLib - Super Library
Output
    *iyIncP - 'yi' of incident particle
    *nZAIncP - ZA of incident particle
    *AWIncP - atomic weight of incident particle
    *xLevIncP - excitation energy of incident particle
    *maxEGP - number of energy groups
    *jGisoP - number of isotopes
    *nProjUP - number of projectiles
    *nYnoP - number of particle descriptors, y
    *iSrcCtrP - source mode control for all particles
        its value is EMISSION if all particles are emission
        its value is REACTION if all particles are reaction
        otherwise its value is MIXED
    *LibTypeP - library type
        1 -multi-group
        2 -fixed grid energy dependent
    (the following four interpolation types are defined:
        0 - undefined
        1 - histogram
        2 - lin-lin
        3 - log-log )
    *ITSigP - cross section interpolation type
    *ITNuP - nubar interpolation type
    *ITMultP - multiplicity interpolation type
    *ITEDepP - energy deposition interpolation type
    retval - return value (0 if no error)

*/

```

**Dat1AccC**

```

int Dat1AccC(ACC *cmd, int *numVal, SuperLibrary *SupLib, void *Val,
             int *numRetVal)
/*
Description
    access one-dimensional data
    (This subroutine must not be called prior to calling 'gen2000'.)
Input
    *cmd      - command integer specifying what to access
                0 ACC_nZAGiso - ZA numbers of found targets
                1 ACC_nRS     - number of reactions by isotope
                2 ACC_AWIso   - atomic weight by isotope
                3 ACC_xLevIso - excitation energy by isotope
                4 ACC_EGBnd   - energy group boundaries by energy group
                5 ACC_iyProj  - 'yo' by projectile
                6 ACC_AWProj  - atomic weight of projectiles
                7 ACC_nZAPProj - ZA numbers of projectiles
                8 ACC_nZAiyi  - ZA numbers of projectiles by iyi
                (values defined in header file)
    *numVal    - length of 'Val'
    *SupLib   - Super Library
Output
    *Val       - array of values accessed
                note: 'Val', 'iVal' and 'xVal' are equivalenced
    *numRetVal - number of elements returned

    retval     - return value
                0 - no error
                1 - size of 'Val' ('*numVal') not large enough
                7 - cmd not available
*/

```

**DistEinSecRAccC**

```

int DistEinSecRAccC(ACCDIST *cmd, int *jiso, int *IrctReq, int *nSP,
                     int *numVal, int *nEinc,
                     SuperLibrary *SupLib,
                     void *Val, int *numRetVal)
/*
Description
    For a specified isotope, reaction, secondary, and incident energy
    index access
    [(nOut,nEincident),(secondary,reaction),iso] data [real]
    (This subroutine must not be called prior to calling 'gen2000'.)
Input
    *cmd      - command integer specifying what to access
                ACC_EOutEP - equal probable bin arrays of energy dists
                ACC_AOutEP - equal probable bin arrays of angular dists
                ACC_AEOutEP- equal probable bin arrays of energy dists as
                function of angle
                (values defined in header file)
    *jiso     - internal isotope number
                (position of isotope in generated library (one-based))
    *IrctReq - requested internal (isotope dependent) reaction number,
                i.e., position of reaction in isotope's list
                (one-based)
    *nSP      - secondary particle index (one-based)
    *nEinc   - incident energy index (one-based)
    *numVal   - number of things (length of 'Val') (currently not used)
    *SupLib   - Super Library
Output
    *Val       - array of values accessed
                note: 'Val', 'iVal' and 'xVal' are equivalenced

```

```

    *numRetVal- number of elements returned

    retval - return value
        0 - no error
    not used -> 1 - size of 'Val' ('*numVal') not large enough
                  2 - reaction '*IrctReq' not found
                  3 - '*nSP' exceeds number of secondaries
                  7 - cmd not available
                  8 - '**jiso' exceeds number of isotopes
*/

```

**DistEinSecRLibAccC**

```

int DistEinSecRLibAccC(ACCDIST *cmd, int *jiso, int *IrctReq, int *nSP,
                      int *numVal, int *nEinc,
                      SuperLibrary *SupLib,
                      void *Val, int *numRetVal)
/*
Description
For a specified isotope, reaction, secondary, and incident energy
index access
(as tabulated in library, i.e., COM frame for kinematics types 1/4/5)
[(nOut,nEincident),(secondary,reaction),iso] data [real]
(This subroutine must not be called prior to calling 'gen2000'.)

Input
*cmd      - command integer specifying what to access
            ACC_EOutEP - equal probable bin arrays of energy dists
            ACC_AOutEP - equal probable bin arrays of angular dists
            ACC_AEOutEP- equal probable bin arrays of energy dists as
                          function of angle
                          (values defined in header file)
*jiso     - internal isotope number
            (position of isotope in generated library (one-based))
*IrctReq - requested internal (isotope dependent) reaction number,
            i.e., position of reaction in isotope's list (one-
            based)
*nSP      - secondary particle index (one-based)
*nEinc   - incident energy index (one-based)
*numVal  - number of things (length of 'Val') (currently not used)
*SupLib   - Super Library

Output
Val       - array of values accessed
            note: 'Val', 'iVal' and 'xVal' are equivalenced
*numRetVal- number of elements returned
retval    - return value
        0 - no error
not used -> 1 - size of 'Val' ('*numVal') not large enough
                  2 - reaction '*IrctReq' not found
                  3 - '*nSP' exceeds number of secondaries
                  7 - cmd not available
                  8 - '**jiso' exceeds number of isotopes
*/

```

**EgpSecRACsBc**

```

int EgpSecRACsBc(ACCESR *cmd, int *jiso, int *IrctReq, int *iySec,
                   int *maxEG, SuperLibrary *SupLib, int *nEGloP,
                   int *nEGhiP, float *xVal)
/*
Description
For a specified isotope, reaction, and secondary, access
[Egroup,(secondary,reaction),iso] data [float]
(This subroutine must not be called prior to calling 'gen2000'.)

Input

```

```

*cmd      - command integer specifying what to access
  x 0 ACC_EDptMult - energy dependent multiplicities
    (values defined in header file)
*jiso     - internal isotope number
    (position of isotope in generated library)
*IrctReq  - requested internal (isotope dependent)
    reaction number
*iySec    - secondary index (may be 'yo' number)
*maxEG    - number of energy groups
*SupLib   - Super Library
Output
  *nEGloP  - first group with 'Mult'
  *nEGhiP  - last group with 'Mult'
  *xVal    - array of values accessed
  retval   - return value
    0 - no error
    7 - cmd not available
    8 - '*jiso' exceeds number of isotopes
*/

```

### **EinSecRAccC**

```

int EinSecRAccC(ACCEIN *cmd, int *jiso, int *IrctReq, int *nSP,
                 int *numVal, SuperLibrary *SupLib, void *Val,
                 int *numRetVal)
/*
Description
  For a specified isotope, reaction, and secondary access
  [nEincident,(secondary,reaction),iso] data [real]
  (This subroutine must not be called prior to calling 'gen2000'.)
Input
  *cmd      - command integer specifying what to access
    ACC_eIETabl   - table of values of incident energy for
                    energy distributions (spectra)
    ACC_eIATabl   - table of values of incident energy for
                    angular distributions
                    (values defined in header file)
  *jiso     - internal isotope number
    (position of isotope in generated library (one-based))
  *IrctReq  - requested internal (isotope dependent) reaction number,
    i.e., position of reaction in isotope''s list (one-
    based)
  *nSP      - secondary particle index (one-based)
  *numVal   - number of things (length of 'Val') (currently not used)
  *SupLib   - Super Library
Output
  *Val      - array of values accessed
    note: 'Val', 'iVal' and 'xVal' are equivalenced
  *numRetVal- number of elements returned
    (if=0 and cmd=ACC_eIATabl then angular distribution is isotropic)
  retval   - return value
    0 - no error
  not used -> 1 - size of 'Val' ('*numVal') not large enough
    2 - reaction '*IrctReq' not found
    3 - '*nSP' exceeds number of secondaries
    7 - cmd not available
    8 - '*jiso' exceeds number of isotopes
*/

```

**Iso2AccC**

```

int Iso2AccC(ACCISO *cmd, int *jiso, int *numVal,
              SuperLibrary *SupLib, int *nZA, void *Val, int *numRetVal)
/*
Description
  For a specified isotope access all things of thing-wise data
  where thing might be reaction, as in nCrct(reaction,iso), or
  energy group, as in STiso(E,iso):
    x(thing,iso)
      (This subroutine must not be called prior to calling 'gen2000')
Input
  *cmd      - command integer specifying what to access
  x 0 ACC_nCrct   - C-number (reaction,iso)
  x 1 ACC_nSrct   - S-number (reaction,iso)
  x 2 ACC_Q0rct   - mass-balance Q (reaction,iso)
                    (if energy dependent, return 999.99
                     for value and the offending reaction
                     index times -1 in 'nZA1')
  x 3 ACC_Ethrct   - reaction threshold (reaction,iso)
  x 4 ACC_KinTyp   - kinematic type (reaction,iso)
  x 5 ACC_nSecProj- number of secondary particles
                    that are projectiles (reaction,iso)
  x 6 ACC_iSecProj- 'yo's of secondary particles
                    that are projectiles
                    [(secondary,reaction),iso]
  x 7 ACC_nProd    - number of "mass balance" particles
                    (reaction,iso)
  x 8 ACC_AW2Bdy   - atomic weight of (1) first particle
                    (2) residual
                    [(1/2,reaction),iso]
  x 9 ACC_xLev2Bdy- excitation of (1) first particle
                    (2) residual
                    [(1/2,reaction),iso]
  x 10 ACC_STiso   - total microscopic cross section (E,iso)
  x 11 ACC_nRct    - number of interaction reactions (E,iso)
  x 12 ACC_nRctTot - total number of reactions (E,iso)
  x 13 ACC_Edep*   - energy local deposition averaged
                    over all reactions (Mev-barn) (E,iso)
  x 14 ACC_Eprod*  - energy production averaged
                    over all reactions (Mev-barn) (E,iso)
  x 15 ACC_gain*   -- gain (particle production) averaged
                    over all reactions (#-barn) (E,iso)
                    (if particle is same as incident gain=nu-1, else
                     gain=nu)
  x 16 ACC_Mult*   - number of each secondary particle
                    that are projectiles
                    (return only integer, energy-independent
                     multiplicities)
                    [(secondary,reaction),iso]
  x 17 ACC_LnFFCoh - number of coherent form factors
                    (reaction,iso)
  x 18 ACC_LnFFInC - number of incoherent form factors
                    (reaction,iso)
  x 19 ACC_gain1
  x 20 ACC_gain2
  x 21 ACC_gain3
  x 22 ACC_gain4
  x 23 ACC_gain5
  x 24 ACC_gain6
  x 25 ACC_gain7
    (values defined in header file)
* constructed from library data
*numVal     - length of 'Val'

```

```

*jiso      - internal isotope number (one-based)
           (position of isotope in generated library)
*numVal   - number of things (length of 'Val')
*SupLib   - Super Library
Output
  *nZA     - ZA number of requested isotope
           (special case for energy-dependent Q0, see above)
  *Val     - array of values accessed
           note: 'Val', 'iVal' and 'xVal' are equivalenced
  *numRetVal - number of elements returned

  retval   - return value
  0 - no error
  1 - size of 'Val' ('*numVal') not large enough
  8 - '*jiso' exceeds number of isotopes
*/

```

**PrdIsoAccC**

```

int PrdIsoAccC(int *jiso, int *numVal, SuperLibrary *SupLib,
                void *Val, int *numRetVal)
/*
Description
  for a specified isotope access all products

Input
  *jiso      - internal isotope number
  *numVal   - number of things (length of 'Val')
  *SupLib   - Super Library
Output
  *Val     - array of values accessed
           note: 'Val', 'iVal' and 'xVal' are equivalenced
  *numRetVal - number of elements returned

  retval   - return value
  0 - no error
  8 - '*jiso' exceeds number of isotopes
*/

```

**PrdRctAccC**

```

int PrdRctAccC(ACCPRD *cmd, int *jiso, int *IrctReq, int *numVal,
                 SuperLibrary *SupLib,
                 int *nZA, void *Val, int *numRetVal)
/*
Description
    for a specified isotope access all products
    for a specified isotope and reaction access all products
        of product-wise data:
            x[product,reaction,iso]
Input
    *cmd      - command integer specifying what to access
    ACC_nZAProd - ZA number of products
    ACC_AWProd - atomic weight of products
    ACC_FFCohx - dependent variable of coherent form factor
    ACC_FFCohS - square of coherent form factor
    ACC_FFCohC - cumulative integral of square
                  of coherent form factor times dependent
                  variable
    ACC_FFInCx - dependent variable of incoherent form factor
    ACC_FFInCy - independent variable of incoherent form
                  factor
                  (values defined in header file)
    *jiso     - internal isotope number
                  (position of isotope in generated library (one-based))
    *IrctReq - requested internal (isotope dependent) reaction number,
                  i.e., position of reaction in isotope's list
                  (one-based)
    *numVal   - number of things (length of 'Val')
    *SupLib   - Super Library
Output
    *nZA      - ZA number of requested isotope
                  (special case for energy-dependent Q0, see above)
    *Val      - array of values accessed
                  note: 'Val', 'iVal' and 'xVal' are equivalenced
    *numRetVal - number of elements returned
    retval    - return value
                0 - no error
                1 - size of 'Val' ('*numVal') not large enough
                7 - cmd not available
                8 - '*jiso' exceeds number of isotopes
*/

```

**RctGrpAccC**

```

int RctGrpAccC(ACCRCT *cmd, int *jiso, int *IrctReq, int *numVal,
                 SuperLibrary *SupLib,
                 int *nZA, int *nEGlo, int *nEGhi, void *Val,
                 int *numRetVal)
/*
Description
    For a specified isotope and reaction access all groups
    of group-wise data that are similar to microscopic reaction
    cross section:
        x[reaction,(E,iso)]
    (This subroutine must not be called prior to calling 'gen2000'.)
Input
    *cmd      - command integer specifying what to access
    ACC_Srct   - microscopic reaction cross section
    ACC_Erct   - local energy deposition by reaction
    ACC_Eprodrt - energy production by reaction (MeV-barn)
    ACC_gainrct - gain (neutron production) by reaction
                  (nu-1)(#-barn)

```

```

ACC_Q0rctEdepndt- energy-dependent mass-balance Q
    (if Q is indeed energy-independent, this
     constant value will be returned as a
     function of group)
    (values defined in header file)
        * constructed from library data
*numVal      - length of 'Val'
*jiso        - internal isotope number
              (position of isotope in generated library (one-based))
*IrctReq    - requested internal (isotope dependent) reaction number,
              i.e., position of reaction in isotope's list
              (one-based)
*numVal      - number of things (length of 'Val')
*SupLib      - Super Library
Output
    *nZA        - ZA number of requested isotope
                  (special case for energy-dependent Q0, see above)
    *nEGlo      - first group with data
    *nEGhi      - last group with data
    *Val         - array of values accessed
                  note: 'Val', 'iVal' and 'xVal' are equivalenced
    *numRetVal- number of elements returned
    retval       - return value
                  0 - no error
                  1 - size of 'Val' ('*numVal') not large enough
                  2 - reaction '*IrctReq' not found
                  7 - cmd not available
                  8 - '*jiso' exceeds number of isotopes
*/

```

**stCalc**

```

void stCalc(const int *nEG, const float *ePart,
            const int *nLiso, const float *xNiso, const int *iLGiso,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            float *st)
/*
Description
    calculate macroscopic total cross section for isotope mixture
    for an incident particle
Input
    *nEG        - energy group of incident particle
    *ePart      - energy of incoming particle
    *nLiso      - number of local isotopes
    *xNiso(j)   - number density of local isotope 'j' (atom/barn-cm)
    *iLGiso(j)  - global internal isotope number ('i' [one-based]
                  corresponding to local internal isotope number 'j'
    *dummynProjU - number of projectiles to track
    *dummymaxEG* - number of energy groups
    *SupLib     - Super Library
    *dummyLenSupLi - only in argument list so that array dimensions
                     will be known
Output
    *st         - macroscopic total cross section for isotope mixture
*/

```

**Collision Routines****bang2000**

```

void bang2000(const int *nEG, const float *ePart, const int *nLiso,
               const float *xNiso, const int *iLGiso,

```

```

        const float *fMultSec,
        float *st, const int *dummynProjU, const int *dummymaxEG,
        SuperLibrary *SupLib, const int *dummyLenSupLi,
        const int *maxYUnq, const int *maxProd,
        int *iLiso, int *iGiso, int *inull,
        int *mrctCol, int *INrctCol, int *nCCol, int *nSCol,
        float *Q0Col,
        int *nProdCol, int *nZAPCol,
        int *nSecYUnqP, int *iSecYUnqP, int *mSecYUnqP,
        int *nSecY, int *iSecY, float *EOut, float *VOut,
        float *AngOut,
        int *ierr, int *dummyioout,
        double (*userrng)(void *), void *rngstate)
/*
Description
    for single incident particle sample
    - isotope
    - reaction
    - output particles
Input
    *nEG      - energy group of incident particle
    *ePart    - energy of incoming particle
    *nLiso    - number of local isotopes
    *xNiso[j] - number density of local isotope 'j' (atom/barn-cm)
                (units are arbitrary as long as all entries in array
                 are consistent with themselves and 'st')
    *iLGiso[j] - global internal isotope number ('i') corresponding
                  to local internal isotope number 'j'
    *fMultSec - multiplier on secondary particle multiplicity
                  given for each projectile in the order given
                  compressed list 'iyProj'.
    *st       - macroscopic total cross section for isotope mixture
                  (if st .le. 0.0 calculate here)
    *dummynProjU* - number of projectiles to track
    *dummymaxEG* - number of energy groups
    *SupLib   - Super Library
    *dummyLenSupLi* - length of 'SupLib'
    *maxYUnq* - size of unique secondary particle lists
    *maxProd* - maximum number of secondary particles,
                  only in argument list so that array dimensions
                  will be known
    *inull    - null reaction counter (input and output)
    *dummyioout - output unit (currently not used)
    *userrng  - user supplied random number generator
    *rngstate - random number generator state
Output
    *iLiso    - local internal number of selected isotope
    *iGiso    - global internal number of selected isotope
    *inull    - null reaction counter (input and output)
    *mrctCol - position in reaction/isotope dependent list of
                  sampled reactions
    *INrctCol - internal reaction number (isotope dependent)
    *nCCol    - reaction descriptor, C
    *nSCol    - reaction switch, S
    *Q0Col    - mass balance Q0 of reaction
    *nProdCol - number of mass balance products
    *nZAPCol - ZA-numbers of mass balance products
                note: mass balance products do not include particles
                      that are being tracked (see 'iyProj' and 'iSecY').
    *nSecYUnqP - number of unique secondary particles
    *iSecYUnqP - list (y numbers) of unique secondary particles
    *mSecYUnqP - list of number of each unique secondary particles
                  (direct access, i.e., mSecYUnqP(iy) = number of type
                   iy, in contrast to mSecYUnq where
                   mSecYUnq(n) = number of type iSecYUnq(n))

```

```

*nSecY      - total number of secondary particles
*iSecY      - list (y numbers) of all secondary particles
              (some may be repeated)
*EOut       - list of energies of secondary particles
*VOut       - list of velocities of secondary particles
*AngOut     - list of cosines of secondary particles
              (rel. to. incident dir.)
* ierr      - error code:
              0 - no error
              1 - not enough isotope cross sections to make total
              2 - not enough reaction cross sections to make total
              3 - passed and computed macroscopic total cross section
                  <= 0.0
*/

```

**SmpIso**

```

void SmpIso(const int *nEG, const float *ePart,
            const int *nLiso, const float *xNiso, const int *iLGiso,
            float *st,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            int *iLiso, int *iGiso, int *inull,
            int *ierr, const int *dummyioout,
            double (*userrng)(void *), void *rngstate)
/*
Description
    for single incident particle sample
    - isotope
Input
    *nEG      - energy group of incident particle
    *ePart    - energy of incoming particle
    *nLiso    - number of local isotopes
    xNiso(j) - number density of local isotope 'j' (atom/barn-cm)
    iLGiso(j) - global internal isotope number ('i' [one-based])
                corresponding to local internal isotope number 'j'
    *st       - macroscopic total cross section for isotope mixture
                (if st .le. 0.0 calculate here)
    *dummynProjU* - number of projectiles to track
    *dummymaxEG* - number of energy groups
    *SupLib   - Super Library
    *dummyLenSupLi* -
    *inull    - null reaction counter
    *dummyioout - output unit
    *userrng  - user supplied random number generator
    *rngstate - random number generator state
    * only in argument list so that array dimensions will be known
Output
    *iLiso    - local internal number of selected isotope [one-based]
    *iGiso    - global internal number of selected isotope [one-based]
    *inull    - null reaction counter
    *ierr      - error code:
                0 - no error
                1 - not enough isotope cross sections to make total
                3 - passed and computed macroscopic total cross section
                    <= 0.0
*/

```

**SmpIsRc**

```

void SmpIsRc(const int *nEG, const float *ePart,
              const int *nLiso, const float *xNiso, const int *iLGiso,
              float *st,
              const int *dummynProjU, const int *dummymaxEG,

```

```

        const SuperLibrary *SupLib, const int *dummyLenSupLi,
        int *iLiso, int *iGiso, int *inull,
        int *mrctCol, int *INrctCol, int *nCCol, int *nSCol,
        float *Q0Col,
        int *nProdCol, int *nZAPCol,
        float *STisoT, float *pfrac,
        int *ierr, const int *dummyioout,
        double (*userrng)(void *), void *rngstate)
/*
Description
    for single incident particle sample
    - isotope
    - reaction
Input
    *nEG      - energy group of incident particle
    *ePart    - energy of incoming particle
    *nLiso    - number of local isotopes
    *xNiso(j) - number density of local isotope 'j' (atom/barn-cm)
    *iLGiso(j)- global internal isotope number ('i') corresponding
                 to local internal isotope number 'j'
    *st       - macroscopic total cross section for isotope mixture
                 (if st .le. 0.0 calculate here)
    *dummynProjU* - number of projectiles to track
    *dummymaxEG* - number of energy groups
    *SupLib   - Super Library
    *dummyLenSupLi*- length of 'SupLib'
                 'SupLib' is a "Super" library that contains
                 'MstrCntl', Master Control section
                 'xMisc', miscellaneous data section
                 'kOfxCntl', offsets of cross section control arrays
                 'ixCtlAr', space occupied by cross section control
                 arrays
                 'xLib', cross section data library
                 'KinCntl', space occupied by kinematic offsets
                 'xKinLib', kinematics data library
    *inull    - null reaction counter*
    *ioout    - output unit
    *userrng  - user supplied random number generator
    *rngstate - random number generator state
    *         - only in argument list so that array dimensions will be known
Output
    *iLiso    - local internal number of selected isotope
    *iGiso    - global internal number of selected isotope
    *inull    - null reaction counter
    *mrctCol - position in reaction/isotope dependent list of
               sampled reactions
    *INrctCol - internal reaction number (isotope dependent)
    *nCCol    - reaction descriptor, C
    *nSCol    - reaction switch, S
    *Q0Col    - mass balance Q0 of reaction
    *nProdCol - number of mass balance products
    *nZAPCol - ZA-numbers of mass balance products
               note: mass balance products do not include particles
                     that are being tracked (see 'iyProj' and 'iSecY').
    *STisoT   - temporarily calculated value of total microscopic
               cross section for 'iGiso'. Valuable for energy-
               dependent libraries.
    *pfrac    - relative position of particle energy in energy
               interval
    *ierr     - error code:
               0 - no error
               1 - not enough isotope cross sections to make total
               2 - not enough reaction cross sections to make total
               3 - passed and computed macroscopic total cross section
                   <= 0.0

```

```
*/
```

### SmpRct

```
void SmpRct(const int *nEG, const float *ePart,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            const int *iGiso, int *inull,
            int *mrctCol, int *INrctCol, int *nCCol, int *nSCol,
            float *Q0Col,
            int *nProdCol, int *NZAPCol,
            float *STisoT, float *pfrac,
            int *ierr, const int *dummyioout,
            double (*userrng)(void *), void *rngstate)
/*
Description
    for single incident particle sample
    - reaction
Input
    *nEG      - energy group of incident particle
    *ePart    - energy of incoming particle
    *dummynProjU* -
    *dummymaxEG* - number of energy groups
    *SupLib   - Super Library
    *dummyLenSupLi* - length of 'SupLib'
    *iGiso    - global internal number of selected isotope [one-based]
    *inull    - null reaction counter
    *ierr     - error code:
                0 - no error
                1 - not enough isotope cross sections to make total
    *dummyioout - output unit
    *userrng   - user supplied random number generator
    *rngstate  - random number generator state
    * only in argument list so that array dimensions will be known
Output
    *inull    - null reaction counter
    *mrctCol  - position in reaction/isotope dependent list of
                sampled reactions [one-based]
    *INrctCol - internal reaction number (isotope dependent)
                [one-based]
    *nCCol    - reaction descriptor, C
    *nSCol    - reaction switch, S
    *Q0Col    - mass balance Q0 of reaction
    *nProdCol - number of mass balance products
    *NZAPCol  - ZA-numbers of mass balance products
                note: mass balance products do not include particles
                that are being tracked (see 'iyProj' and 'iSecY').
    *STisoT   - temporarily calculated value of total microscopic
                cross section for 'iGiso'. Valuable for energy-
                dependent libraries.
    *pfrac    - relative position of particle energy in energy interval
    *ierr     - error code:
                0 - no error
                1 - not enough isotope cross sections to make total
                2 - not enough reaction cross sections to make total
*/
```

### SmpRctMB

```
void SmpRctMB(const int *nEG, const float *ePart,
               const int *dummynProjU, const int *dummymaxEG,
               const SuperLibrary *SupLib, const int *dummyLenSupLi,
               const int *iGiso, int *inull,
               int *mrctCol, int *INrctCol, int *nCCol, int *nSCol,
```

```

        float *Q0Col,
        int *nProdCol, int *nZAPCol,
        int *ierr, const int *dummyioout,
        double (*userrng)(void *), void *rngstate)
/*
Description
for single incident particle sample
- reaction
Special Version for Multi-Band
exclude reactions C=10,15,46

Input
*nEG      - energy group of incident particle
*ePart     - energy of incoming particle
*dummynProjU*
*dummymaxEG*   - number of energy groups

*SupLib    - Super Library
*dummyLenSupLi*- length of 'SupLib'
*iGiso     - global internal number of selected isotope [one-based]
*inull     - null reaction counter
*ierr       - error code:
            0 - no error
            1 - not enough isotope cross sections to make total
*dummyioout - output unit
*userrng    - user supplied random number generator
*rngstate   - random number generator state
* only in argument list so that array dimensions will be known
Output
*inull     - null reaction counter
*mrctCol   - position in reaction/isotope dependent list of
            sampled reactions [one-based]
*INrctCol - internal reaction number (isotope dependent)
            [one-based]
*nCCol     - reaction descriptor, C
*nSCol     - reaction switch, S
*Q0Col     - mass balance Q0 of reaction
*nProdCol  - number of mass balance products
*nZAPCol   - ZA-numbers of mass balance products
            note: mass balance products do not include particles
            that are being tracked (see 'iyProj' and 'iSecY').
*ierr       - error code:
            0 - no error
            1 - not enough isotope cross sections to make total
            2 - not enough reaction cross sections to make total
*/

```

**SmpSrc**

```

void SmpSrc(const int *nEG, const float *ePart, const int *INrctCol,
            const float *fMultSec,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            const int *maxYUnq, const int *maxProd,
            float *STIsoT, const float *pfrac,
            const int *iGiso, int *inull,
            int *nSecYUnqP, int *iSecYUnqP, int *mSecYUnqP,
            int *nSecY, int *iSecY, float *EOut, float *VOut,
            float *AngOut,
            int *ierr, const int *dummyioout,
            double (*userrng)(void *), void *rngstate)

```

```

/*
Description
    for single incident particle sample
        - output particles
Input
    *nEG      - energy group of incident particle
    *ePart    - energy of incoming particle
    *INrctCol - internal reaction number (isotope dependent)
                [one-based]
    *fMultSec - multiplier on secondary particle multiplicity
                given for each projectile in the order given
                compressed list 'iyProj'
    *dummynProjU* - number of projectiles to track
    *dummymaxEG* - number of energy groups
    *SupLib   - Super Library
    *dummyLenSupLi*- length of 'SupLib'
    *maxYUnq - size of unique secondary particle lists
    *maxProd - maximum number of secondary particles
    *STisoT   - temporarily calculated value of total microscopic
                cross section for 'iGiso'. Valuable for energy-
                dependent libraries.
                (if .le. 0.0 this routine will calculate)
    *pfrac    - relative position of particle energy in energy interval
* only in argument list so that array dimensions will be known
    *inull    - null reaction counter
    *ierr     - error code:
                0 - no error
                1 - not enough isotope cross sections to make total
                2 - not enough reaction cross sections to make total
                3 - number of source particles exceeds space allocated
                    (nSecY .gt. maxProd)
    *iGiso    - global internal number of selected isotope [one-based]
    *dummyioout - output unit
    *userrng  - user supplied random number generator
    *rngstate - random number generator state
Output
    *inull    - null reaction counter
    *nSecYUnqP - number of unique secondary particles
    *iSecYUnqP - list (y numbers) of unique secondary particles
    *mSecYUnqP - list of number of each unique secondary particles
                (direct access, i.e., mSecYUnqP(iy-1) = number of
                type iy, in contrast to mSecYUnq where
                mSecYUnq(n) = number of type iSecYUnq(n))
                (In Fortran calling routine this is a 1-based array,
                thus 'iy-1')
    *nSecY    - total number of secondary particles
    *iSecY    - list (y numbers) of all secondary particles
                (some may be repeated)
    *EOut     - list of energies of secondary particles
    *VOut     - list of velocities of secondary particles
    *AngOut   - list of cosines of secondary particles
                (rel. to. incident dir.)
*/

```

## Client Customization Routines

### **ElimReact**

```

void ElimReact(const int * const jiso, const int * const IrctReq,
               const SuperLibrary * const SupLib,
               int *nZA, int *nC, int *nS, int *ierr)

```

```
/*
Description
  For a single isotope, eliminate the requested reaction.
  (This function must not be called prior to calling 'gen2000'.)
Input
  *jiso      - internal global isotope number
               (position of isotope in generated library (one-based))
  *IrctReq   - requested internal (isotope dependent) reaction number,
               i.e., position of reaction in isotope's list
               (one based)
  *SupLib    - Super Library
Output
  *nZA       - ZA number of modified isotope
  *nC        - eliminated reaction descriptor, 'C' number
  *nS        - eliminated reaction switch, 'S' number
  *ierr      - error flag
               0 - no error
               1 - '*jiso' exceeds number of available isotopes
               2 - '*IrctReq' exceeds number of available reactions
*/
```

**XSecModC**

```

int XSecModC(const XSMOD * const cmd, const float * const xModVal,
             const int * const jiso, const int * const IrctReq,
             const int * const nEGlo, const int * const nEGhi,
             const SuperLibrary * const SupLib,
             int *nZA, int *nC, int *nS)

/*
Description
For a specified isotope, reaction, and energy group range,
modify (multiply, add to, or set equal to a specified value)
the microscopic reaction cross section data as determined by
the argument cmd. The groupwise total cross section is
recomputed in each case.
(This function must not be called prior to calling 'gen2000'.)

```

## NOTE 1:

Only the routines XSecMult (fortran) and XSecMultC (C) can be used to modify microscopic reaction cross section data during SuperLibrary generation. Only XSecMod (fortran) and XSecModC (C) can be used to modify microscopic reaction cross section data after the SuperLibrary has been generated.

## NOTE 2:

Passing xModVal[n] == 0.0, for any n, into XSecMod or XSecModC is not allowed for iyIncP == 7 (gamma) and ITSig == LOGLOG because the cross section data is stored and interpolated in log-log space. Passing xModVal[n] == 0.0 under these conditions results in an error being returned. The ElimReact function can be used instead.

**Input**

|          |  |
|----------|--|
| *cmd     | - microscopic reaction cross section modification command  |
|          | x 0 XSMOD_EIndptMult - multiply the cross section by<br>energy-independent *xModVal  |
|          | x 1 XSMOD_EDptMult - multiply the cross section by<br>energy-dependent xModVal[n]  |
|          | x 2 XSMOD_EIndptAdd - add the energy-independent<br>*xModVal to cross section  |
|          | x 3 XSMOD_EDptAdd - add the energy-dependent<br>xModVal[n] to cross section  |
|          | x 4 XSMOD_EIndptSet - set the cross section equal to<br>energy-independent *xModVal  |
|          | x 5 XSMOD_EDptSet - set the cross section equal to<br>energy-dependent xModVal[n]  |
| *xModVal | - value by which to modify microscopic cross section<br>(If *cmd==XSMOD_EIndptMult, *cmd==XSMOD_EIndptAdd<br>or *cmd==XSMOD_EIndptSet, then the scalar *xModVal<br>is used to modify all energy groups in the specified<br>range. If *cmd==XSMOD_EDptMult, *cmd==XSMOD_EDptAdd,<br>or *cmd==XSMOD_EDptSet, then xModVal must be an array<br>of length *nEGhi - *nEGlo + 1 whose elements are used<br>to modify the cross section.) |
| *jiso    | - internal global isotope number<br>(position of isotope in generated library (one-based))   |
| *IrctReq | - requested internal (isotope dependent) reaction number,<br>i.e., position of reaction in isotope's list<br>(one-based)   |
| *nEGlo   | - lowest energy group to modify (one-based)  |
| *nEGhi   | - highest energy group to modify (one-based)   |
| *SupLib  | - Super Library  |

**Output**

```

*nZA      - ZA number of modified isotope
*nC       - reaction descriptor, 'C' number, of modified
            cross section
*ns       - reaction switch, 'S' number, of modified cross section

retval   - return value
          0 - no error
          1 - '*jiso' exceeds number of available isotopes
          2 - '*IrctReq' exceeds number of available reactions
          3 - energy group range not valid
          4 - reaction '*IrctReq' not found in energy group range
          7 - fatal
              modification option not available
          8 - fatal
              *xModVal<=0.0 for iyIncP==7 and ITSig==LOGLOG
          9 - fatal
              cross section after modification is negative
*/

```

**XSecMultC**

```

float XSecMultC (int iyInc, int nZA, int nC, int ns, float Ethresh,
                  int nR)

/*
Description
    Return to 'gen2000' a multiplier to be applied to microscopic
    cross sections so that non-analog physics may be performed.
    Multiplier is applied during generation to _all_ energy groups!!
Input
    iyInc   - y number of incident particle
    nZA     - ZA number of isotope
    nC      - reaction descriptor, 'C' number
    ns      - reaction switch, 'S' number
    Ethresh - reaction threshold
            (useful when above four identifiers fail to
             uniquely identify a reaction, e.g. 5011(n,n''))
    nR      - position of reaction in isotopes reaction list
            (useful when nS=3 (switch for gamma energy), the
             only difference between reactions is the gamma
             energy which is imbedded in the kinematics data and
             is not readily available when 'XSecMult' is called.)
Output
    XSecMultC - cross section multiplier to be applied as
                coded below (returned as value of function)
                always return '1.0' for analog physics
*/

```

**ZASubMisc**

```

void ZASubMiscC(int *nZAProd, float *AWProd, int *nProd, int *nZAGiso,
                 int jGiso, int iyInc, int nZA, int nC, int ns,
                 float Ethresh, int nR)
/*
Description
    replace ZA numbers in product arrays with indices
    and allow isotope substitution if isotope is not available
Input
    nZAProd - ZA number of products
    AWProd   - atomic mass of products
    nProd    - number of "mass balance" products
    nZAGiso - ZA numbers of isotopes in generated library
    jGiso    - number of isotopes
    iyInc   - y number of incident particle

```

```

nZA      - ZA number of this isotope
nC       - reaction descriptor, 'C' number
nS       - reaction switch, 'S' number
Ethresh - reaction threshold
          (useful when above four identifiers fail to
           uniquely identify a reaction, e.g. 5011(n,n''))
nR       - position of reaction in isotopes reaction list
          (useful when nS=3 (switch for gamma energy), the
           only difference between reactions is the gamma
           energy which is imbedded in the kinematics data and
           is not readily available when 'XSecMult' is called.)
Output
  nZAProd - substituted index for product
  AWProd   - atomic mass of products
  nProd    - number of products (may be shortened)
*/

```

**PnInit**

```

void PnInit (const double *an, double *cn, const int *porder,
             double *ppmin, double *ppmax)
/*
Description
  determines min/max of a sum of Legendre polynomials
  f(z)=Sum(an*Pn(z))
  and the coefficients cn in f(z)=Sum(cn*z^n); n=0,1,2...
Input
  *an - coefficient of Legendre polynomial
  *porder - max order of polynomial (<50)
Output
  *cn - coefficient of summed Legendre polynomials
  *ppmin - min of f(z) in [-1,1]
  *ppmax - max of f(z) in [-1,1]
*/

```

**PnSamp**

```

double PnSamp (const double *cn, const int *porder, const double *pmax,
                double (*userrng)(void *), void *rngstate)

/*
Description
  samples polynomial pdf by rejection method
Input
  *cn - coefficients of polynomial; n=0,1,2...
  *porder - order of polynomial
  *pmax - max value of poly in [-1,1]
  *userrng - user supplied random # generator
  *rngstate - generator state
Output
  PnSamp - sample in [-1,1]
*/

```

**MCAPM Include File****mcapm.h**

```

typedef enum {READ, WRITE} pdbMode;
typedef enum { ACC_nZAGiso, ACC_nRS, ACC_AWIso, ACC_xLevIso,
               ACC_EGBnd, ACC_iyProj,
               ACC_AWProj, ACC_nZAPproj, ACC_nZAiyi } ACC;
typedef enum { ACC_nCrct, ACC_nSrct, ACC_Q0rct, ACC_Ethrct,

```

```

ACC_KinTyp, ACC_nSecProj, ACC_iSecProj,
ACC_nProd, ACC_AW2Bdy, ACC_xLev2Bdy,
ACC_STiso, ACC_nRct, ACC_nRctTot, ACC_Edep,
ACC_Eprod, ACC_gain, ACC_Mult, ACC_LnFFCoh,
ACC_LnFFInc, ACC_gain1, ACC_gain2, ACC_gain3,
ACC_gain4, ACC_gain5, ACC_gain6, ACC_gain7} ACCISO;
typedef enum { ACC_Srct, ACC_Erct, ACC_Eprodrt,
               ACC_gainrct, ACC_Q0rctEdepndt } ACCRCT;
typedef enum { ACC_eIETabl, ACC_eIATabl } ACCEIN;
typedef enum { ACC_EOutEP, ACC_AOutEP, ACC_AEOutEP } ACCDIST;
typedef enum { ACC_nZAProd, ACC_AWProd, ACC_FFCohx, ACC_FFCohS,
               ACC_FFCohC, ACC_FFIInCx, ACC_FFIInCy } ACCPRD;
typedef enum { ACC_EDptMult } ACCESR;
typedef enum { XSMOD_EIndptMult, XSMOD_EDptMult,
               XSMOD_EIndptAdd, XSMOD_EDptAdd,
               XSMOD_EIndptSet, XSMOD_EDptSet } XSMOD;
void gen2000(int *iGiso, int *nGiso, int *iTReqd, int *iyProjIn,
             int *nProjU, int *iyIncP, SourceMode *jSrcCtr, int *maxEG,
             int *nCrsFin, float *xFinWght, int *libdaty, int *nZAInc,
             int *NumTmp, int *nYno, LibraryType *LibTyp,
             InterpType *ITSig, InterpType *ITEDep, int *jGiso,
             int *nZAGiso, int *nRS, SuperLibrary **SupLibP,
             int *LenSupLib, int *ierr,
             int *ioLib, int *itoout, int *itoerr);
void bang2000(const int *nEG, const float *ePart, const int *nLiso,
               const float *xNiso, const int *iLGiso,
               const float *fMultSec, float *st, const int *dummynProjU,
               const int *dummymaxEG,
               SuperLibrary *SupLib, const int *dummyLenSupLi,
               const int *maxYUnq, const int *maxProd,
               int *iLiso, int *iGiso, int *inull, int *mrctCol,
               int *INrctCol, int *nCCol, int *nSCol, float *Q0Col,
               int *nProdCol, int *nZAPCol, int *nSecYUnqP,
               int *iSecYUnqP, int *mSecYUnqP, int *nSecY,
               int *iSecY, float *EOut, float *VOut, float *AngOut,
               int *ierr, int *dummyioout,
               double (*userrng)(void *, void *));
void ProIso(int maxEG, int *nCrsFin, float *xFinWght,
            int *iGiso, int nGiso, PDBfile *pdbLibFile,
            int *nZAisoLibFile, int numIsoLibFile,
            SuperLibrary *SupLib);
void ProY(int *nProjU, int *iyProjIn, SourceMode *jSrcCtr,
          int iyIncP, yInfo *yProjInfo, SuperLibrary *SupLib);
int opendump2000(char *pdbDumpFileName, PDBfile **pdbDumpFile,
                  pdbMode mode);
int closedump2000(PDBfile *pdbDumpFile);
int dump2000(SuperLibrary *SupLib, PDBfile *pdbDumpFile);
int undump2000(SuperLibrary **pSupLib, PDBfile *pdbDumpFile);
int Dat0AccC (SuperLibrary *SupLibP,
               int *iyIncP, int *nZAIncP, float *AWIncP,
               float *xLevIncP, int *maxEGx, int *jGisoP, int *nProjUP,
               int *nYnoP, int *iSrcCtrP, int *LibTypeP,
               int *ITSigP, int *ITNuP, int *ITMultP, int *ITEDepP);
int Dat1AccC(ACC *cmd, int *numVal, SuperLibrary *SupLib,
              void *Val, int *numRetVal);
int Iso2AccC(ACCISO *cmd, int *jiso, int *numVal,
              SuperLibrary *SupLib, int *nZA, void *Val,
              int *numRetVal);
int RctGrpAccC(ACCRCT *cmd, int *jiso, int *IrctReq,
                int *numVal, SuperLibrary *SupLib, int *nZA,
                int *nEGlo, int *nEGhi, void *Val, int *numRetVal);
int EinSecRLibAccC(ACCEIN *cmd, int *jiso, int *IrctReq,
                    int *nSP, int *numVal, SuperLibrary *SupLib,
                    void *Val, int *numRetVal);
int DistEinSecRLibAccC(ACCDIST *cmd, int *jiso, int *IrctReq,

```

```

        int *nSP, int *numVal, int *nEinc,
        SuperLibrary *SupLib, void *Val, int *numRetVal);
int EinSecRAccC(ACCEIN *cmd, int *jiso, int *IrctReq,
                int *nSP, int *numVal, SuperLibrary *SupLib,
                void *Val, int *numRetVal);
int DistEinSecRAccC(ACCDIST *cmd, int *jiso, int *IrctReq,
                    int *nSP, int *numVal, int *nEinc,
                    SuperLibrary *SupLib, void *Val, int *numRetVal);
int PrdRctAccC(ACCPRD *cmd, int *jiso, int *IrctReq,
                int *numVal, SuperLibrary *SupLib,
                int *nZA, void *Val, int *numRetVal);
int PrdIsoAccC(int *jiso, int *numVal, SuperLibrary *SupLib,
               void *Val, int *numRetVal);
int XSecModC(const XSMOD * const cmd, const float * const xModVal,
             const int * const jiso, const int * const IrctReq,
             const int * const nEGlo, const int * const nEGhi,
             const SuperLibrary * const SupLib,
             Int *nZA, int *nC, int *nS);
void ElimReact(const int * const jiso, const int * const IrctReq,
               const SuperLibrary * const SupLib,
               int *nZA, int *nC, int *nS, int *ierr);
void PreViewC(int *iyInc, int *libdaty, int *nZAInc, int *MatNum,
              int **nZA, int *NumTmp, float **Tmp, int *maxEG,
              int *ierr, int *ioLib);
void InfoRctC(SuperLibrary *SupLib, int *jiso, int *INrct,
               int *nC, int *nS, float *Q0, int *nProd, int *nZA,
               int *ierr);
void BalancZAC(int *nZAtrg, int *nSecProj, int *iSecProj,
               int *Mult, int *nProd, int *nZAProd,
               int *nZAProj, int *nZAiyi, int *nZAInc,
               int *nZBal, int *nABal, int *nZABal);
void PreVProdsC(int *iyIncP, int *iGiso, int *nGiso,
                 int *libdatyP, int *nProd, int **nZAProd, int *ierr);
void stCalc(const int *nEG, const float *ePart,
            const int *nLiso, const float *xNiso, const int *iLGiso,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            float *st);
void SmpIso(const int *nEG, const float *ePart,
            const int *nLiso, const float *xNiso, const int *iLGiso,
            float *st, const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            int *iLiso, int *iGiso, int *inull,
            int *ierr, const int *dummyioout,
            double (*userrng)(void *), void *);
void SmpRct(const int *nEG, const float *ePart,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            const int *iGiso, int *inull, int *mrctCol,
            int *INrctCol, int *nCCol, int *nSCol, float *Q0Col,
            int *nProdCol, int *nZAPCol, float *STisoT,
            float *pfrac, int *ierr, const int *dummyioout,
            double (*userrng)(void *), void *);
void SmpIsRc(const int *nEG, const float *ePart,
             const int *nLiso, const float *xNiso, const int *iLGiso,
             float *st, const int *dummynProjU, const int *dummymaxEG,
             const SuperLibrary *SupLib, const int *dummyLenSupLi,
             int *iLiso, int *iGiso, int *inull, int *mrctCol,
             int *INrctCol, int *nCCol, int *nSCol, float *Q0Col,
             int *nProdCol, int *nZAPCol, float *STisoT,
             float *pfrac, int *ierr, const int *dummyioout,
             double (*userrng)(void *), void *);
void SmpRctMB(const int *nEG, const float *ePart,
              const int *dummynProjU, const int *dummymaxEG,
              const SuperLibrary *SupLib, const int *dummyLenSupLi,

```

```

        const int *iGiso, int *inull, int *mrctCol,
        int *INrctCol, int *nCCol, int *nSCol, float *Q0Col,
        int *nProdCol, int *nZAPCol, int *ierr,
        const int *dummyioout,
        double (*userrng)(void *), void *);
void SmpSrc(const int *nEG, const float *ePart,
            const int *INrctCol, const float *fMultSec,
            const int *dummynProjU, const int *dummymaxEG,
            const SuperLibrary *SupLib, const int *dummyLenSupLi,
            const int *maxYUnq, const int *maxProd,
            float *STIsoT, const float *pfrac,
            const int *iGiso, int *inull,
            int *nSecYUnqP, int *iSecYUnqP, int *mSecYUnqP,
            int *nSecY, int *iSecY, float *EOut, float *VOut,
            float *AngOut, int *ierr, const int *dummyioout,
            double (*userrng)(void *), void *);
void Rel2BodyC(const float * const xM1s,
               const float * const xM2s, const float * const xM3s,
               const float * const xM4s, const float * const wls,
               const float * const w2s, const float * const w3s,
               const float * const w4s, const float * const Els,
               const float * const COScoms,
               OutParticle * const Out1, OutParticle * const Out2);
void EPInterpC(const float ePart, const EProb * const EP,
               float * const xOutPart, float * const xOutAlph,
               int * const nBin, int * const iEinLo,
               float * const eInAlph,
               double (*userrng)(void *), void *);
void EPInterpLC(const float ePart, const EProb * const EP,
                float * const xOutPart, float * const xOutAlph,
                int * const nBin, int * const iEinLo,
                float * const eInAlph,
                double (*userrng)(void *), void *);
void AEInterpC(int * const nAngBin, int * const iEinLo,
               float * const eInAlph, float * const AOutAlph,
               const AngEng * const AE, float * EOutPart,
               double (*userrng)(void *), void *);
void kinematic(const float * const ePart,
               const int * const iGiso, const int * const INrctCol,
               const int nC, const ParticleList * const Unique,
               const SuperLibrary * const SupLib,
               const int * const maxProd, int * const inull,
               int * const nSecY, int * const iSecY,
               OutParticle * const Out, OutParticle * const Recoil,
               double (*userrng)(void *), void *);
void CohAngleC(const float eGamma, const float * const FFCohx,
               const float * const FFCohS, const float * const FFCohC,
               const int LnFFCoh, float *AOut,
               double (*userrng)(void *), void *);
void LnLnIntC(const int i, const float xval,
               const float * const f,
               const float * const x, float *xIntval);
void LnLnInvC(const int i, const float r,
               const float * const fc, const float * const f,
               const float * const x, float *xInv);
void InCohSmpC(const float eGamma,
               const float * const FFInCx,
               const float * const FFInCy, const int LnFFInC,
               float *AOut, float *eGamOut,
               double (*userrng)(void *), void *);
void kleinC(const float el, float *e2, float *cs,
            double (*userrng)(void *), void *);
void LnLnInpC(const float * const f, const float * const x,
               const int len, float xReq, float *fReq);
void ProFFacC(const float jGiso,

```

```
    const SuperLibrary * const SupLib);
float XSecMultC (int iyInc, int nZA, int nC, int nS,
                 float Ethresh, int nR);
void ZASubMisc(int *nZAProd, float *AWProd, int *nProd,
               int *nZAGiso, int jGiso, int iyInc, int nZA,
               int nC, int nS, float Ethresh, int nR);
void GrpClpsC(int maxEGCrs, int *nCrsFin, float *xFinWght,
               int maxEGFin, float *EGb, float *xValFin,
               float *xFactFin, int LenFin, int nEGloFin,
               int nEGhiFin, float **xValCrs, int *LenCrs,
               int *nEGloCrs, int *nEGhiCrs);
void ProGrpDatC(SuperLibrary *SupLib, IsotopeInfo *Iso,
                  float *EGBndCrs, int maxEG, float **SrctCrs,
                  int *LenSrctCrs, int *nEGloCrs, int *nEGhiCrs,
                  float **ErctCrs);
int EgpSecRAcSbC(ACCESR *cmd, int *jiso, int *IrctReq,
                  int *iySec, int *maxEG, SuperLibrary *SupLib,
                  int *nEGloP, int *nEGhiP, float *xVal);
void OpenLibC(const int iyIncP, PDBfile **pdbLibFile,
              const char *chSubNam, int *ierr);
void GetLibPhC(const char *chSubNam, char *pmcapmPath);
void Getid_mcapm (char *str);
void PnInit (const double *an, double *cn, const int *porder,
             double *ppmin, double *ppmax);
double PnSamp (const double *cn, const int *porder, const double *pmax,
               double (*userrng)(void *), void *rngstate);
```

# Index

- 55, 21
- Access Routines, 17, 18, 19, 20, 25
- accretion, 2, 16
- $\alpha r$ , 22
- atomic weight, 18, 19
- BalancZAC, 17, 18, 25, 45
- bang2000, ii, 1, 2, 3, 7, 16, 17, 21, 23, 33, 44, 50
- Be9(n,2n)2 $\alpha$ , 6
- charged particle, 4, 10, 13
- client customizing, 39
- closed system, 5
- cluster, 6
- C-number, 15, 19
- code units, 15
- coherent scattering, 6
- collision routines, 33
- Compaq, ii, 22
- conservation
  - energy, 5, 6
  - momentum, 5, 6
- continuous energy, 3
- correlation, 21
- cross sections
  - charged particle, 3
  - gamma, 3
  - neutron, 3
- cross sections, 2, 19, 20
- d(n,2n)p, 6
- Dat0AccC, 17, 18, 26, 44
- Dat1AccC, 17, 18, 27, 44
- data library, 22
- decay, 6
- depletion, 5
- DistEinSecRAccC, 17, 19, 27, 45
- DistEinSecRLibAccC, 17, 19, 28, 44
- distributions
  - angle, 4
  - coupled angle-energy, 4
  - energy, 4
- Eg<sub>n</sub>SecRAcSbC, 17, 19, 28, 47
- EinSecRAccC, 17, 19, 29, 45
- ElimReact, 20, 39, 41, 45
- emission, 21
- energy, 2
- energy conservation, 5, 6
- energy deposition, 2, 16, 18, 19, 20, 22, 24, 26, 32
- energy group boundaries, 3, 9, 10, 18, 19
- energy points, 3, 11
- energy production, 19, 20
- energy/angle correlated, 6
- energy/angle uncorrelated, 5
- equal probable bins, 4
- excitation energy, 18, 26, 27
- form factor, 6, 7, 32
- gain, 19, 20
- gamma, 7, 8, 11, 14, 21, 41, 42, 43, 50
- gen2000, ii, 1, 2, 3, 4, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 32, 40, 41, 44, 50
- generator, 23
- IBM, ii, 22
- include files, 43
- incoherent scattering, 7
- interpolation type, 18
- Iso2AccC, 17, 19, 22, 30, 44
- isotopes, 4, 12, 13, 14, 16
- kinematics type, 5
  - type 1, 5
  - type 2, 5
  - type 3, 6
  - type 4, 6
  - type 5, 6
  - type 6, 6
  - type 7, 7
- kinematics type, 19
- Klein-Nishina cross section, 7
- libmcapm.a, 22
- library
  - charged particle, 10, 13
  - file names, 8
  - gamma, 11, 14
  - neutron, 9
  - previewer, 23
- library type, 18
- Linux, ii, 22
- makefile, 22
- mass-balance Q, 19
- mcapm.h, 16, 22, 43
- mcapm.tar, 22
- missing isotopes, 5
- momentum, 2
- momentum conservation, 5, 6
- Monte Carlo statistics, 21
- multi-group, 3, 26
- multiplicities, 4, 19
- neutron, 2, 3, 4, 6, 8, 9, 12, 19, 20, 32
- non-analog physics, 21
- PACT, 22
- particle weights, 21
- pdb, 2, 3, 4, 8, 22

Plechaty Hybrid scheme, 3  
PnInit, 17, 21, 43, 47  
PnSamp, 21, 43, 47  
PrdIsoAccC, 17, 19, 31, 45  
PrdRctAccC, 17, 19, 32, 45  
PreViewC, 16, 17, 18, 23, 45  
*products*, 2, 22  
projectile index, 18  
projectiles, 2, 4, 8, 22  
proton, 2, 6, 8  
random number generator, 22, 34, 35, 36, 37, 38,  
    39  
RCS, 22  
RctGrpAccC, 17, 20, 22, 32, 44  
reaction, 16, 21  
reaction IDs, 15  
relativistic two-body, 6  
sampling  
    reaction, 16  
    source, 16  
    target isotope, 16  
secondary particle, 2  
SGI, ii, 22  
SmpIso, 16, 35, 45  
SmpIsRc, 16, 35, 45  
SmpRct, 16, 37, 45, 50  
SmpRctMB, 16, 37, 45  
SmpSrc, 16, 38, 46, 50  
S-number, 19  
source mode, 18  
sourcing, 16, 21, 24  
space, 21  
stCalc, 17, 20, 33, 45  
Sun, ii, 22  
tar, 22  
Thomson cross section, 6  
threshold, 3, 19  
UNIX, 22  
XSecModC, 20, 41, 45  
XSecMultC, 20, 41, 42, 47  
y-number, 16, 18  
ZA number, 18, 19, 25, 31, 32, 33, 40, 42  
ZASubMisc, 20, 42, 47

## Document Revision History

|          |     |     |   |
|----------|-----|-----|---|
| 11/4/91  | jar |     | Initial revision.   |
| 12/1/91  | jar |     | Initial revision.   |
| 1/20/92  | jar |     | Initial revision.   |
| 1/31/92  | jar |     | Initial revision.   |
| 2/9/92   | jar |     | Initial revision.   |
| 6/19/92  | jar |     | Initial revision.   |
| 1/21/93  | jar | 0.1 | added gamma description<br>added gropu collapse (gen2000)                                       |
| 2/17/93  | jar | 0.2 | changed interface to gen2000, SmpRct, and<br>SmpSrc.  |
| 5/27/93  | jar | 0.3 | add I/O unit description to gen2000 and<br>bang2000<br>add spacer and relspace<br>correct typos |
| 10/16/00 | cah | 1.0 | added documentation describing C version  |